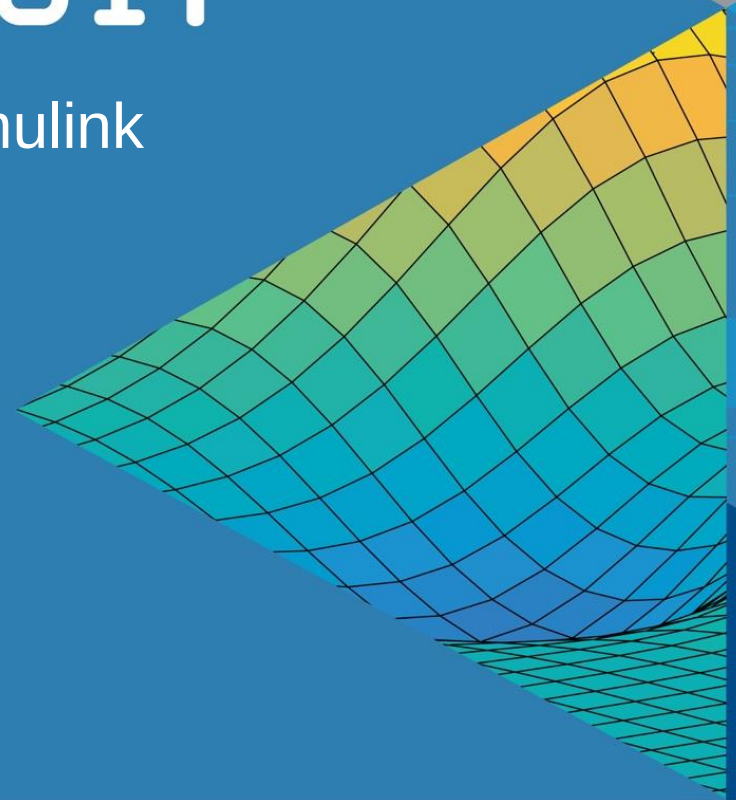
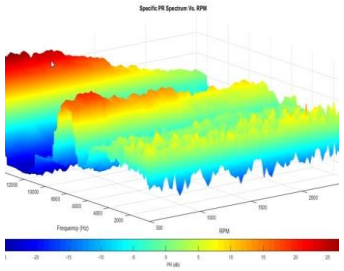
MATLAB EXPO 2017

Parallel Computing with MATLAB and Simulink



Loren Dean, MathWorks



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

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

Design and Build Wave Energy Farm
Sensitivity studies accelerated 12x

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

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

Overcome Challenges

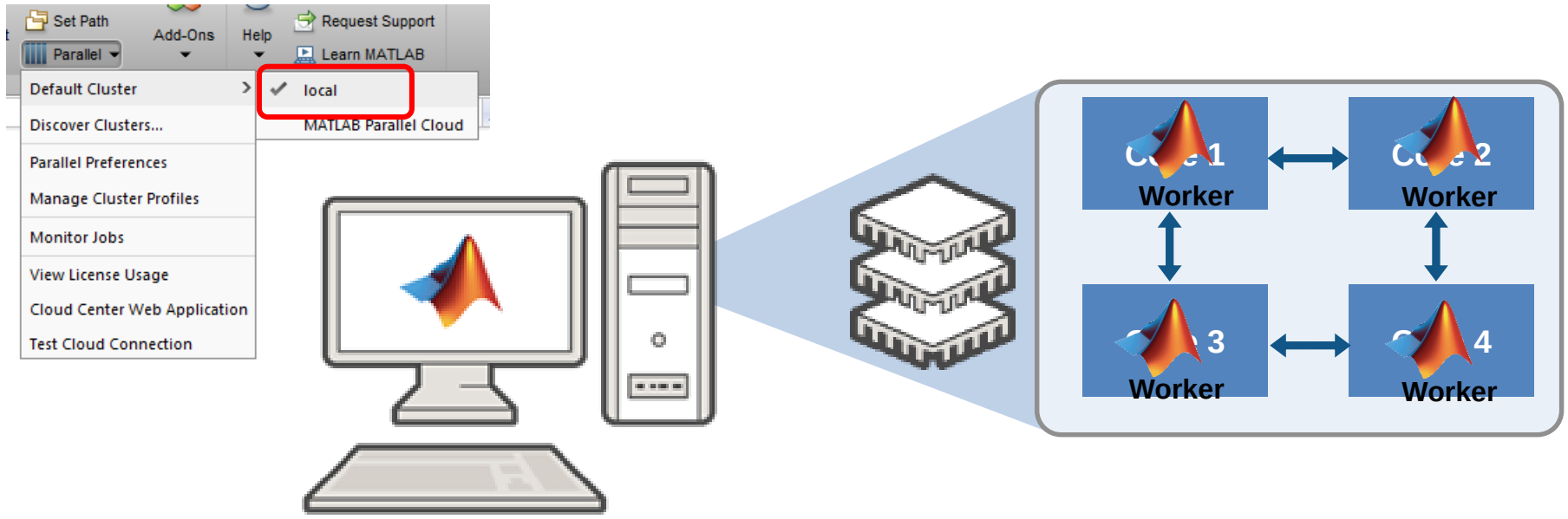
- Challenges
  - Need faster insight to bring competitive products to market quickly
  - Complexity of analytical problems is growing across industries
- Key Takeaways
  - Save engineering and research time and focus on results
  - Leverage computational power of multicore desktops, GPUs, clusters, and clouds
  - Seamlessly scale from your desktop to clusters or the cloud

Agenda

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

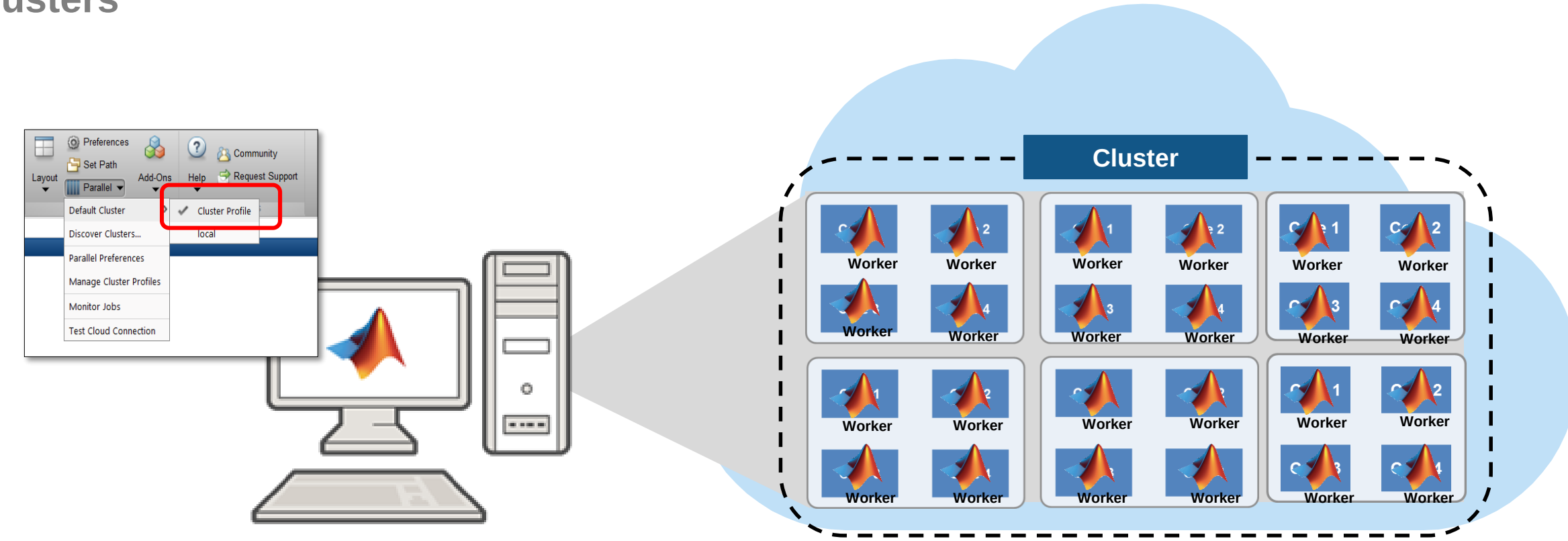
Parallel Computing Paradigm

Multicore Desktops

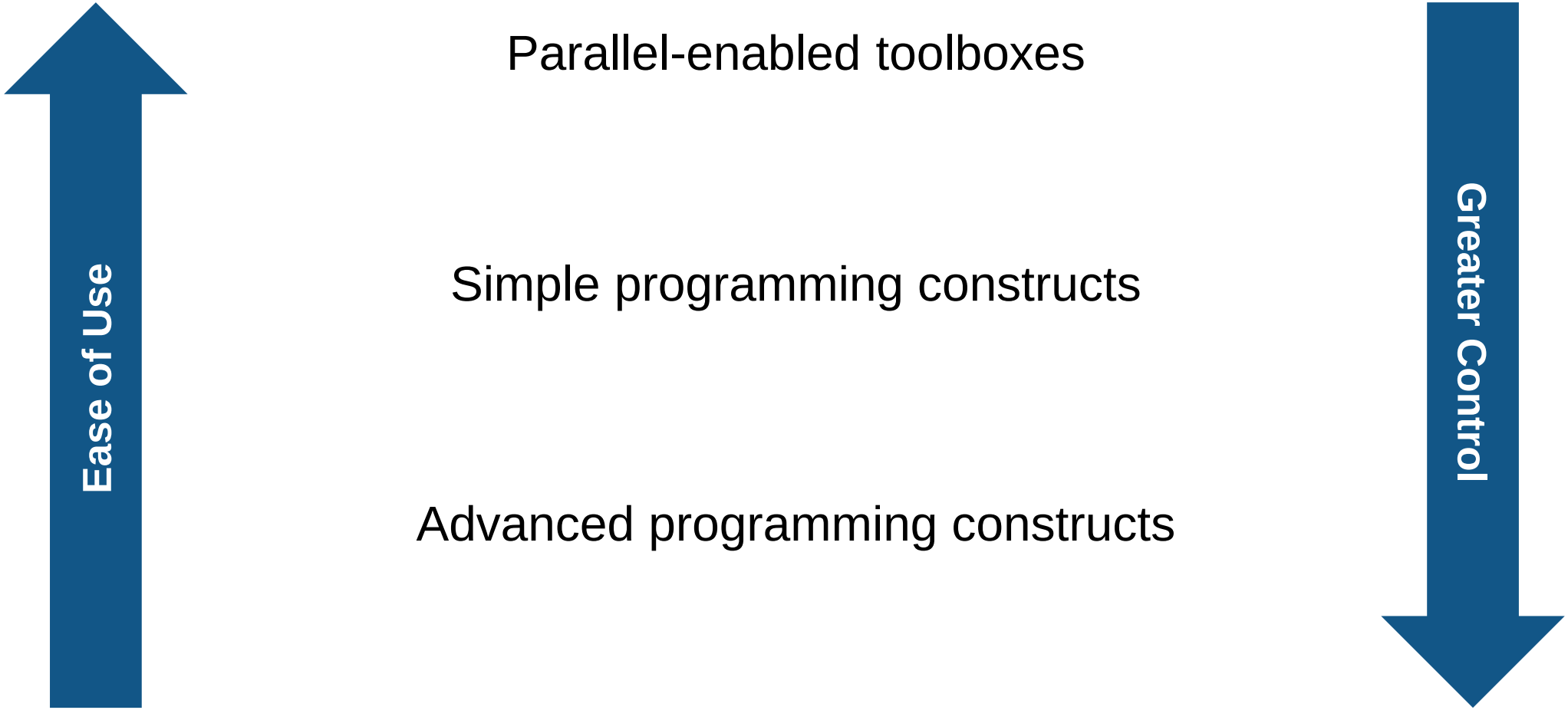


Parallel Computing Paradigm

Clusters



Accelerating MATLAB and Simulink Applications



Parallel-enabled Toolboxes (MATLAB® Product Family)

Enable acceleration by setting a flag or preference

Image Processing

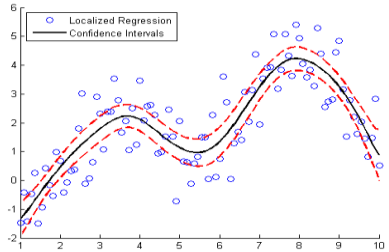
Batch Image Processor, Block Processing, GPU-enabled functions

Original Image of Peppers

Recolored Image of Peppers

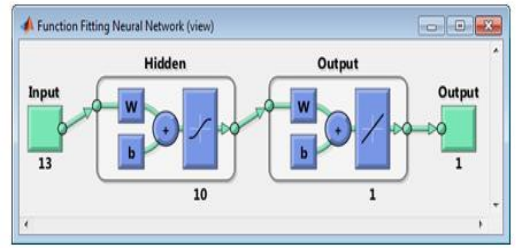
Statistics and Machine Learning

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



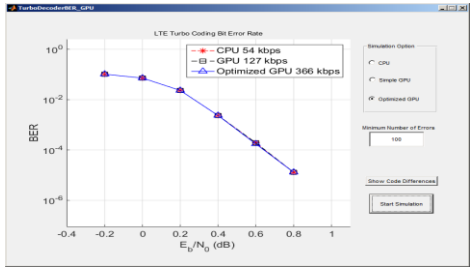
Neural Networks

Deep Learning, Neural Network training and simulation



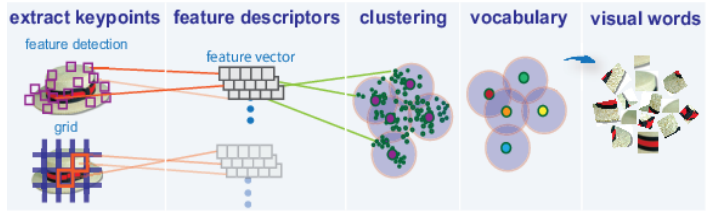
Signal Processing and Communications

GPU-enabled FFT filtering, cross correlation, BER simulations



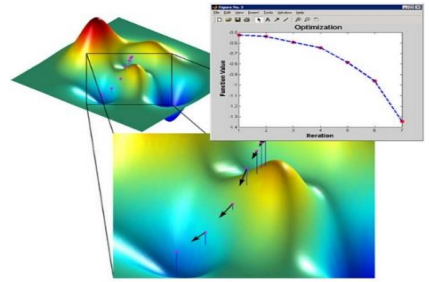
Computer Vision

Bag-of-words workflow



Optimization

Estimation of gradients

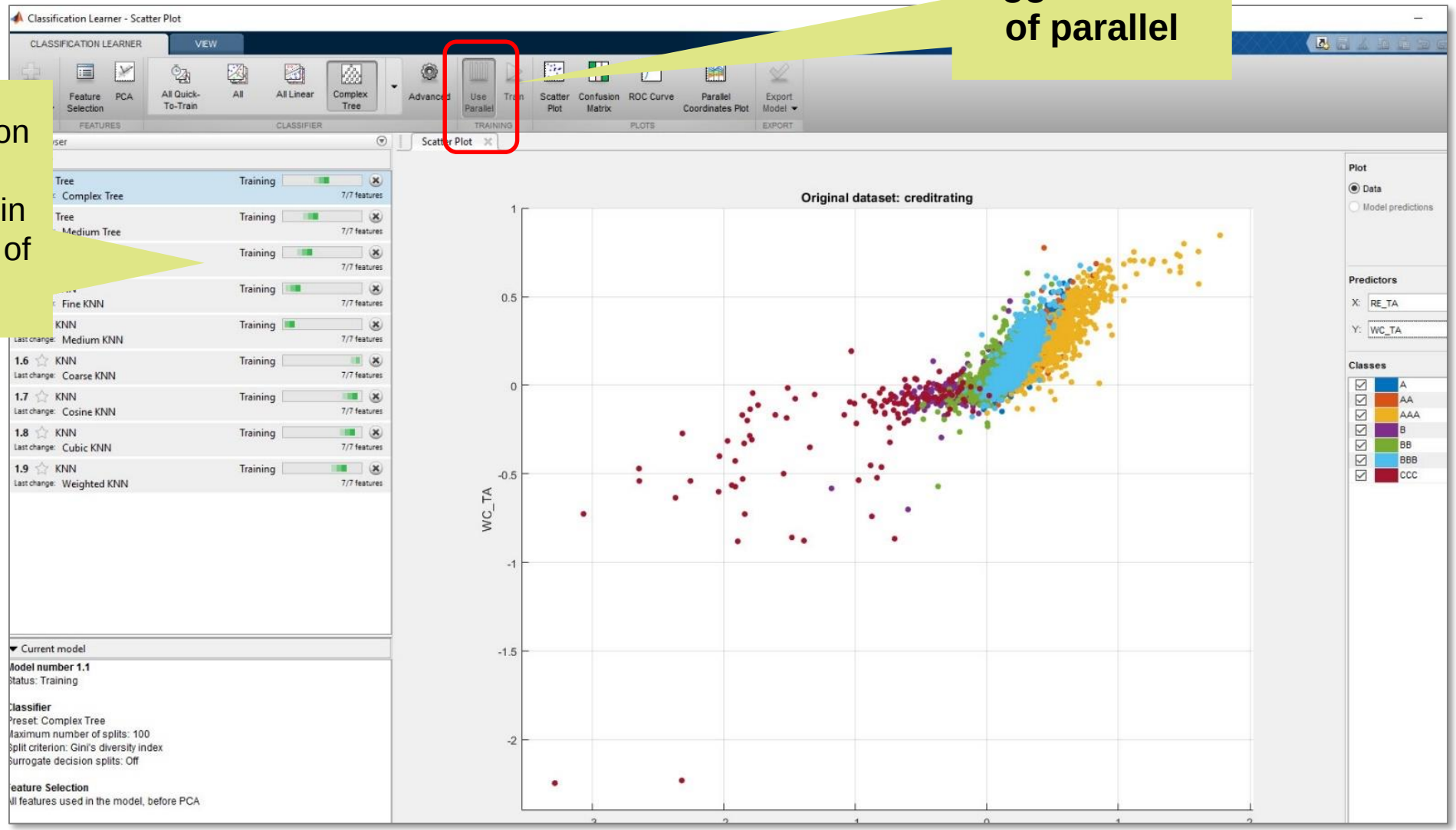


Other Parallel-enabled Toolboxes

Classification learner demo

One click to toggle the use of parallel

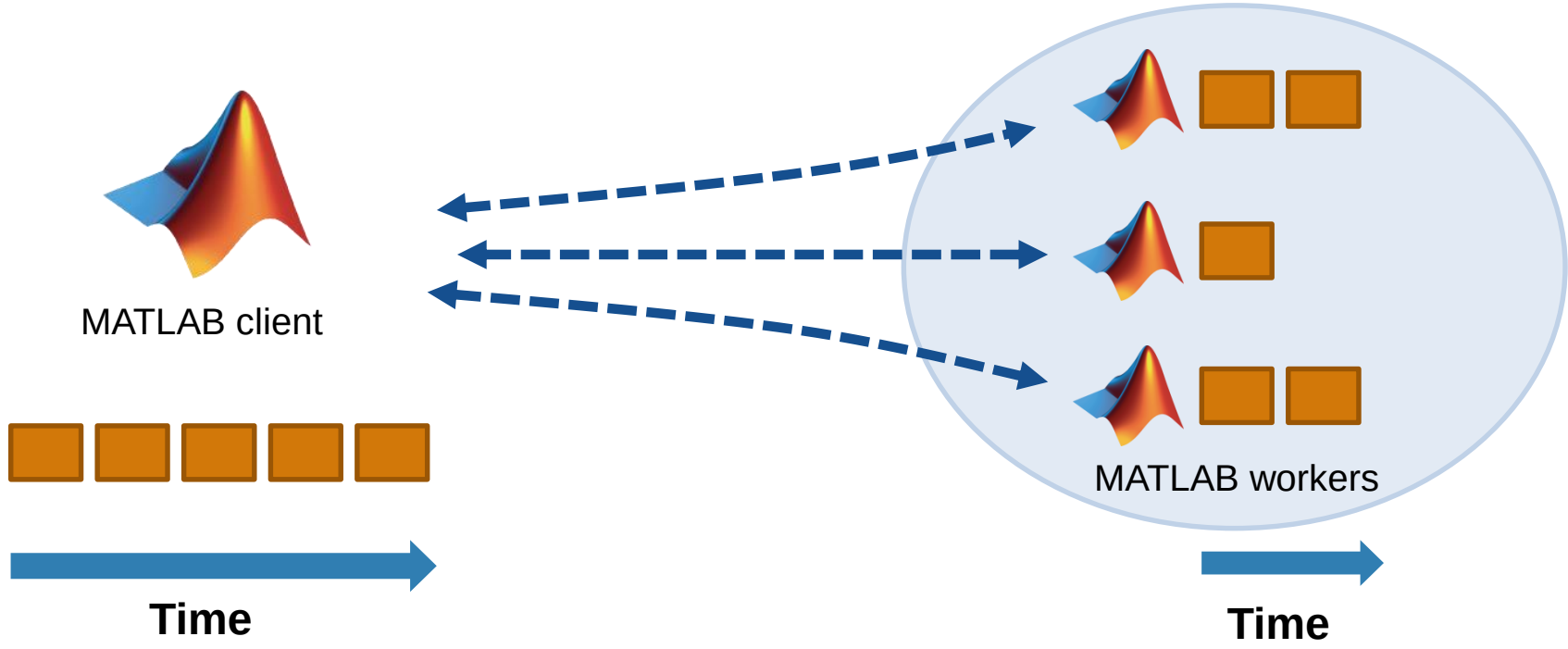
Run classification learner quick to train classifiers in parallel instead of one by one



Explicit Parallelism: Independent Tasks or Iterations

Simple programming constructs: `parfor, parfeval`

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



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****."*

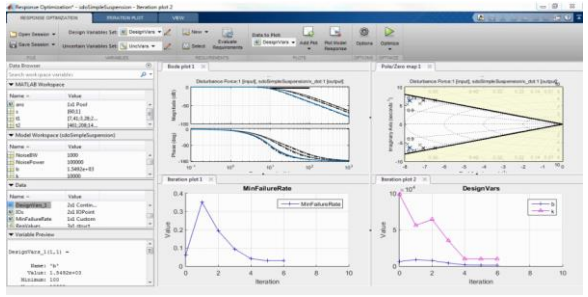
Jonathan Fiévez
Carnegie Wave Energy

Parallel-enabled Toolboxes (Simulink® Product Family)

Enable parallel computing support by setting a flag or preference

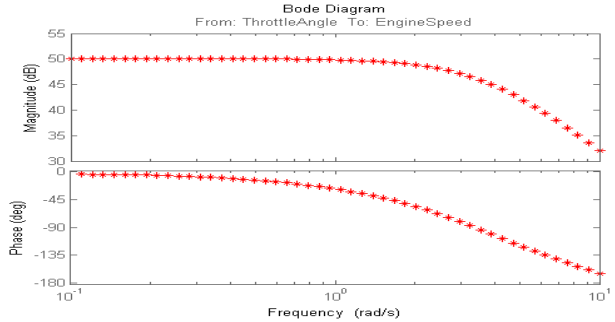
Simulink Design Optimization

Response optimization, sensitivity analysis, parameter estimation



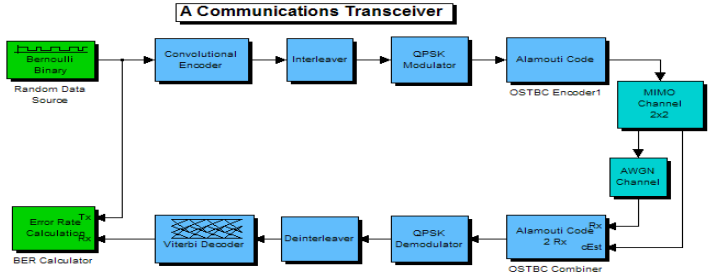
Simulink Control Design

Frequency response estimation



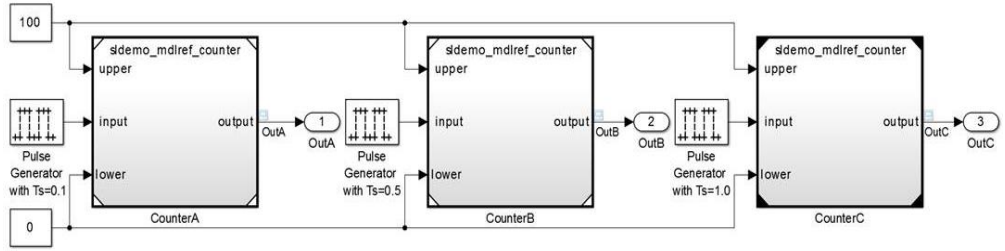
Communication Systems Toolbox

GPU-based System objects for Simulation Acceleration



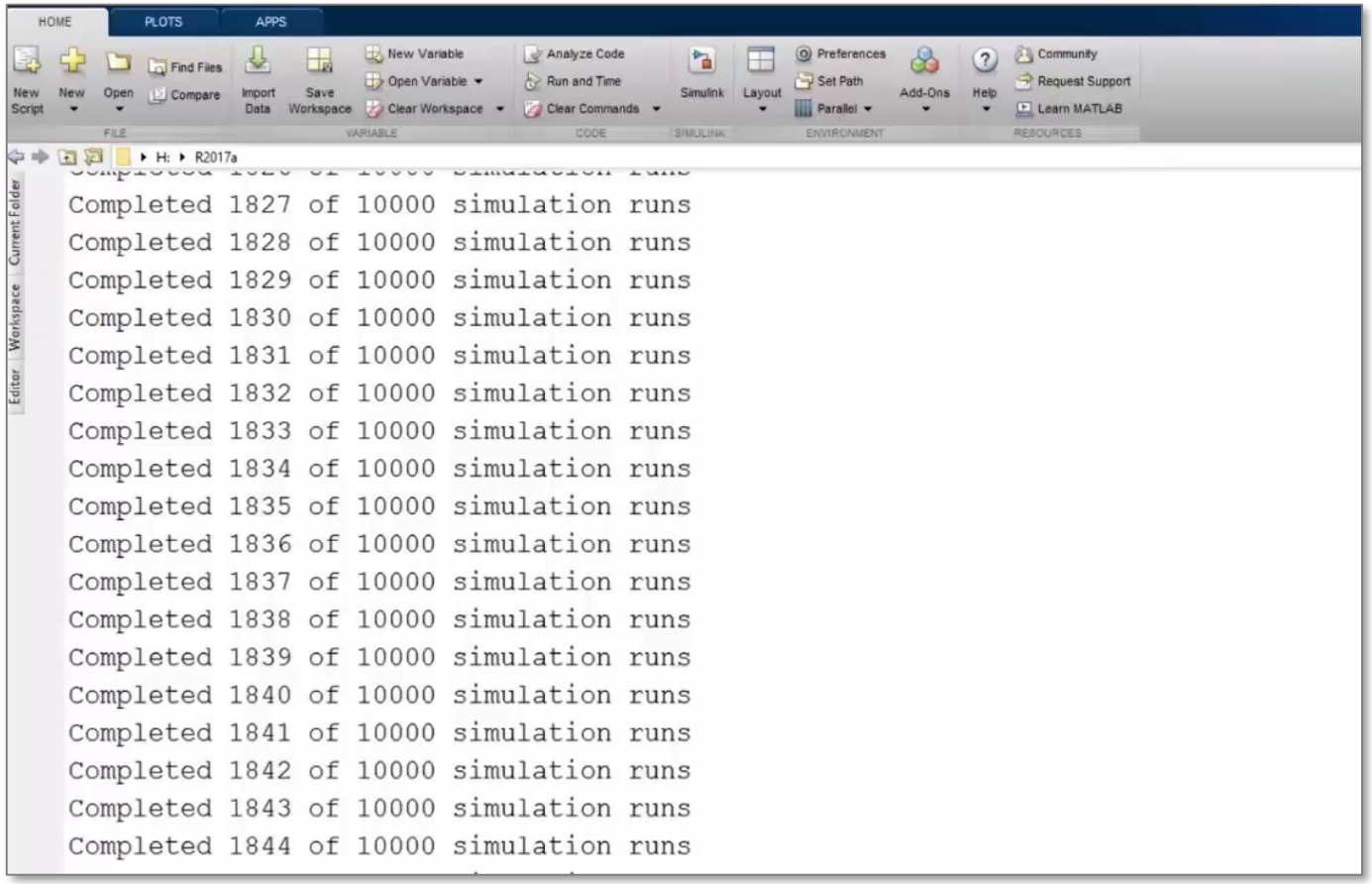
Simulink/Embedded Coder

Generating and building code



[Other Parallel-enabled Toolboxes]

Run multiple parallel simulations from the `parsim` command

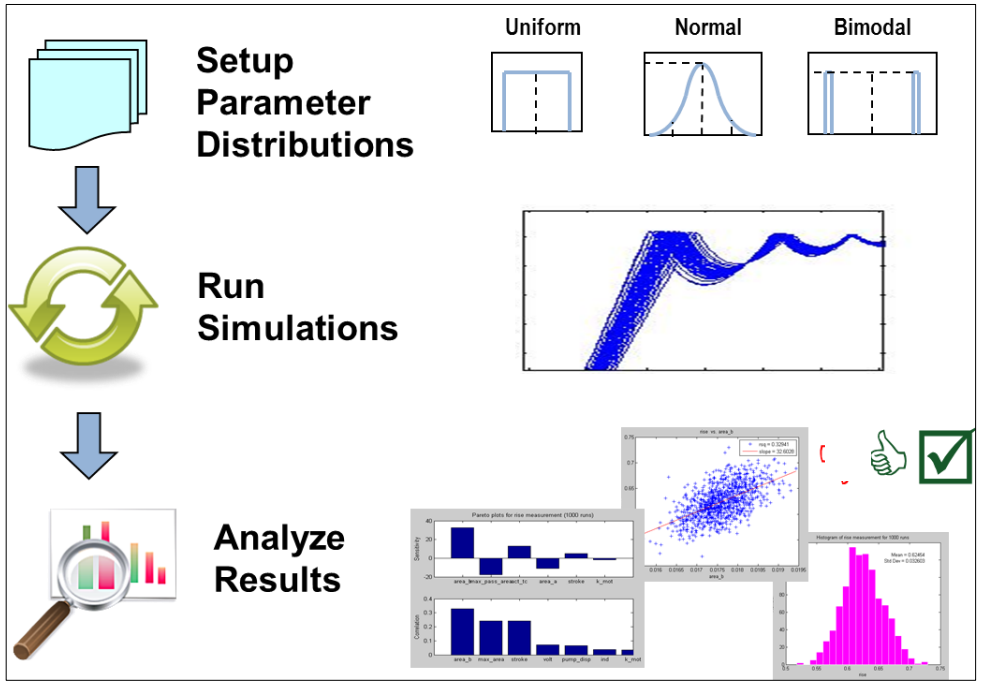


- Enables Simulink users to speed up simulations and simplify workflow
- Simplifies users' large simulation runs and improves their productivity

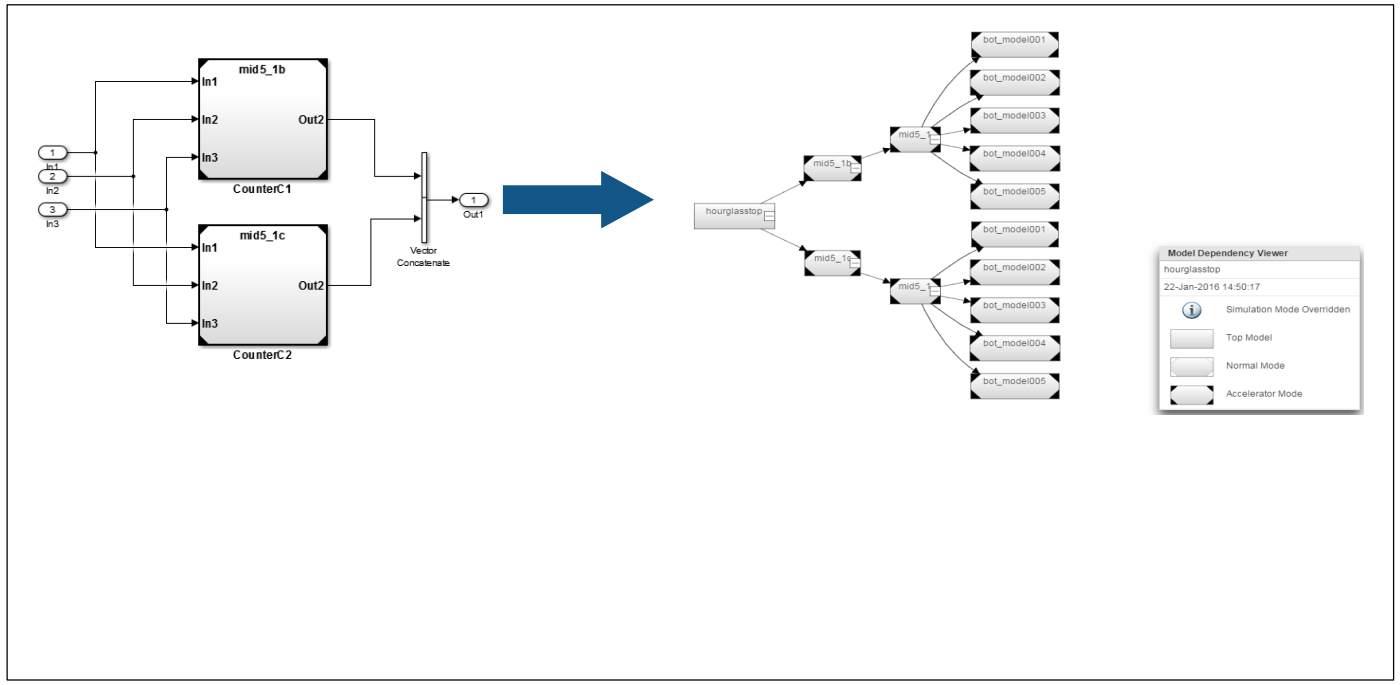
Leverage Parallel Computing with Simulink

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

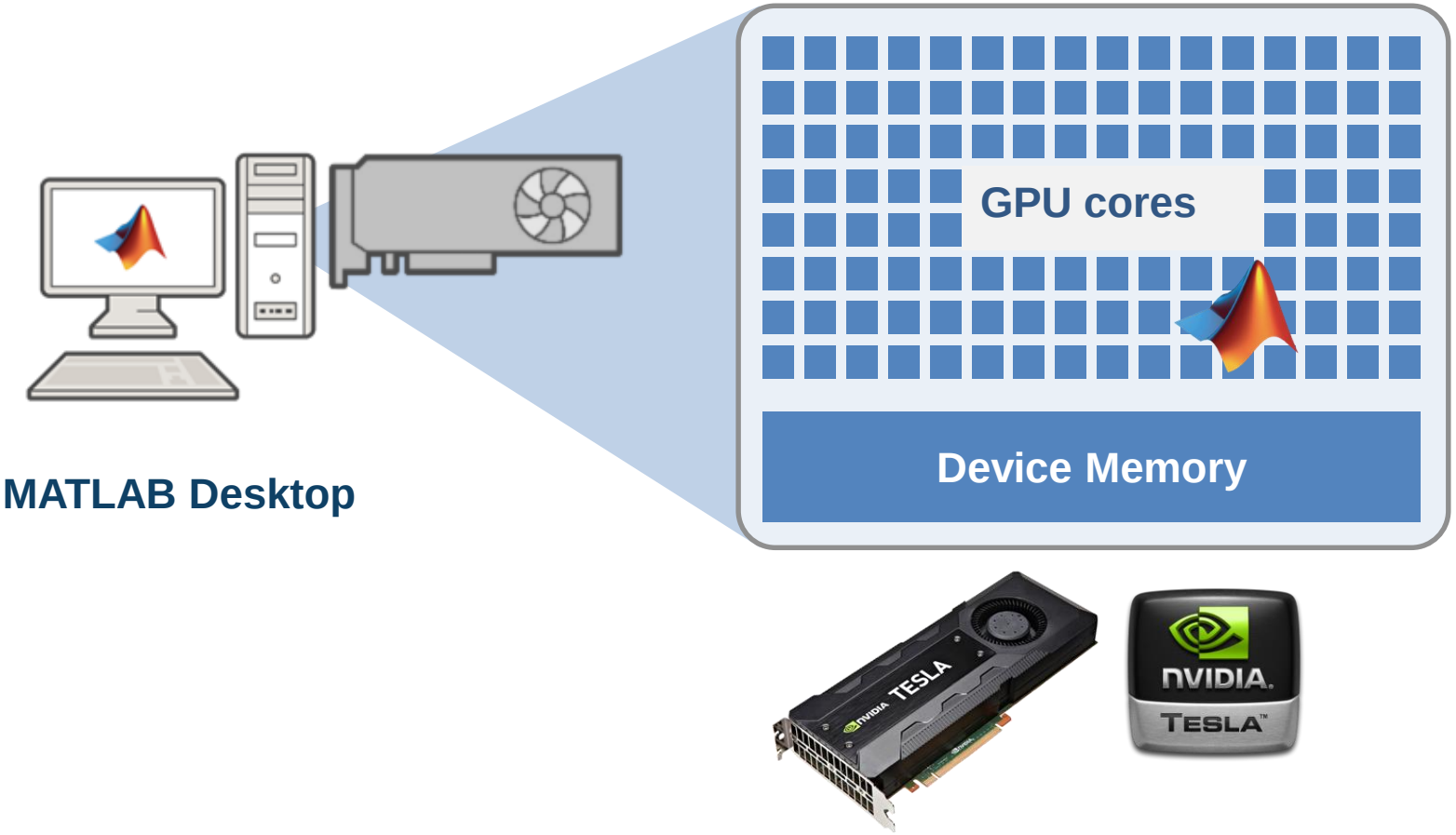


Agenda

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

Parallel Computing Paradigm

NVIDIA GPUs



Speed-up using NVIDIA GPUs

- Ideal Problems
  - Massively Parallel and/or Vectorized operations
  - Computationally Intensive
  - Algorithm consists of supported functions
- 300+ GPU-enabled MATLAB functions
- Additional GPU-enabled Toolboxes
  - Neural Networks
  - Image Processing
  - Communications
  - Signal Processing

Transfer Data To GPU From Computer Memory

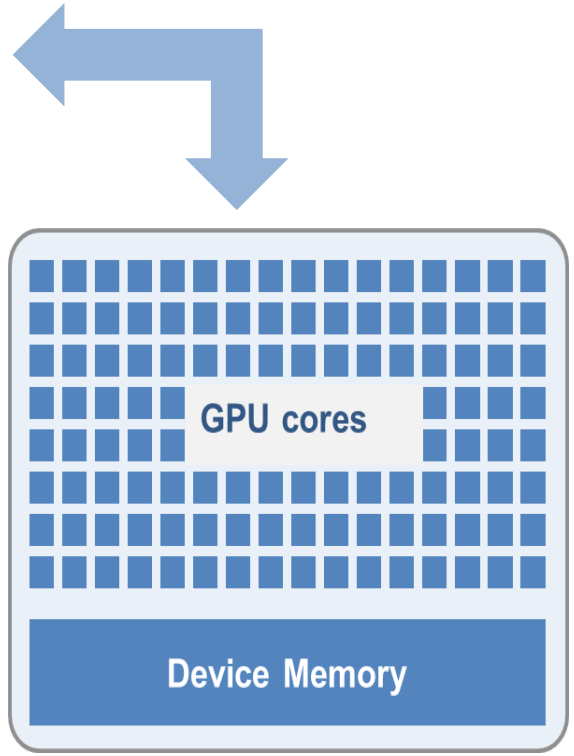
```
A=gpuArray(A);
```

Perform Calculation on GPU

```
X=exping(A);
```

Gather Data or Plot

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



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

Agenda

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

Big Data workflow

ACCESS

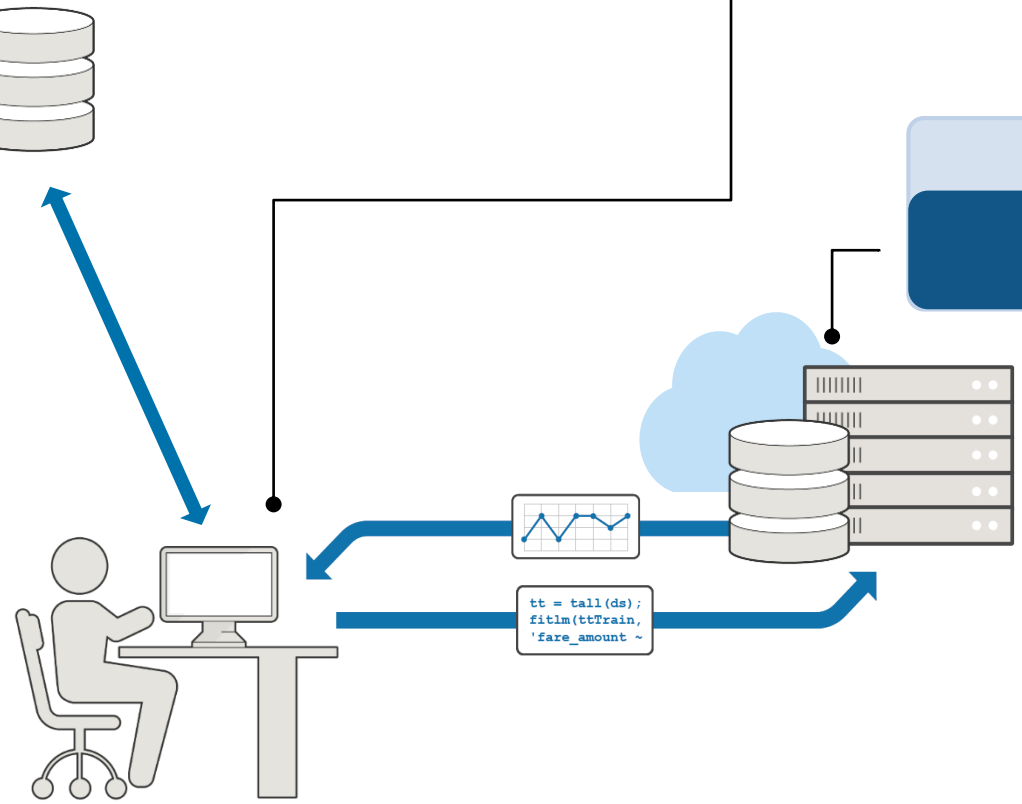
More data and collections of files than fit in memory

PROCESS AND ANALYZE

Adapt traditional processing tools or learn new tools to work with Big Data

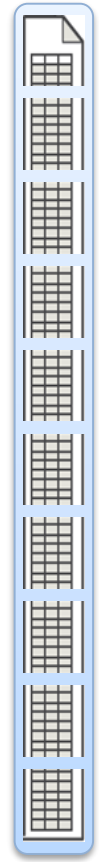
SCALE

To Big Data systems like Hadoop



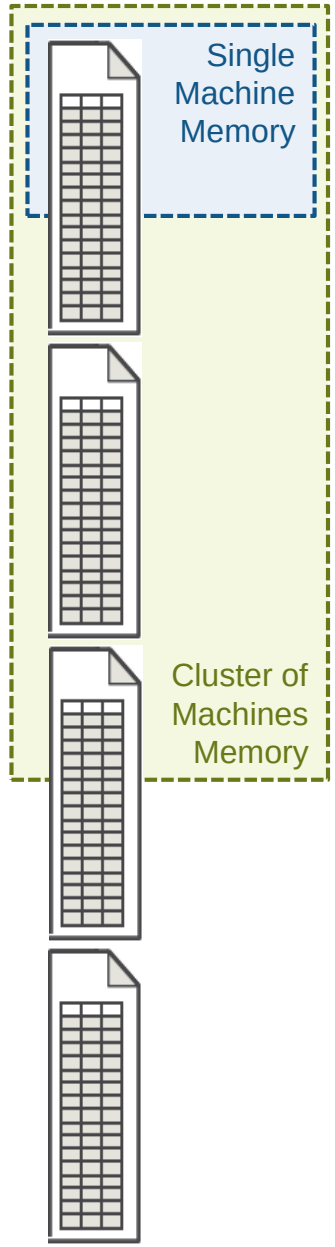
`tall` arrays R2016b

- Data doesn't fit into memory, even cluster memory
- Lots of observations (hence "tall")
- Looks like a normal MATLAB array
  - Numeric types, tables, datetimes, strings, etc…
  - Basic math, stats, indexing, etc.
  - **Statistics and Machine Learning Toolbox**
    (clustering, classification, etc.)



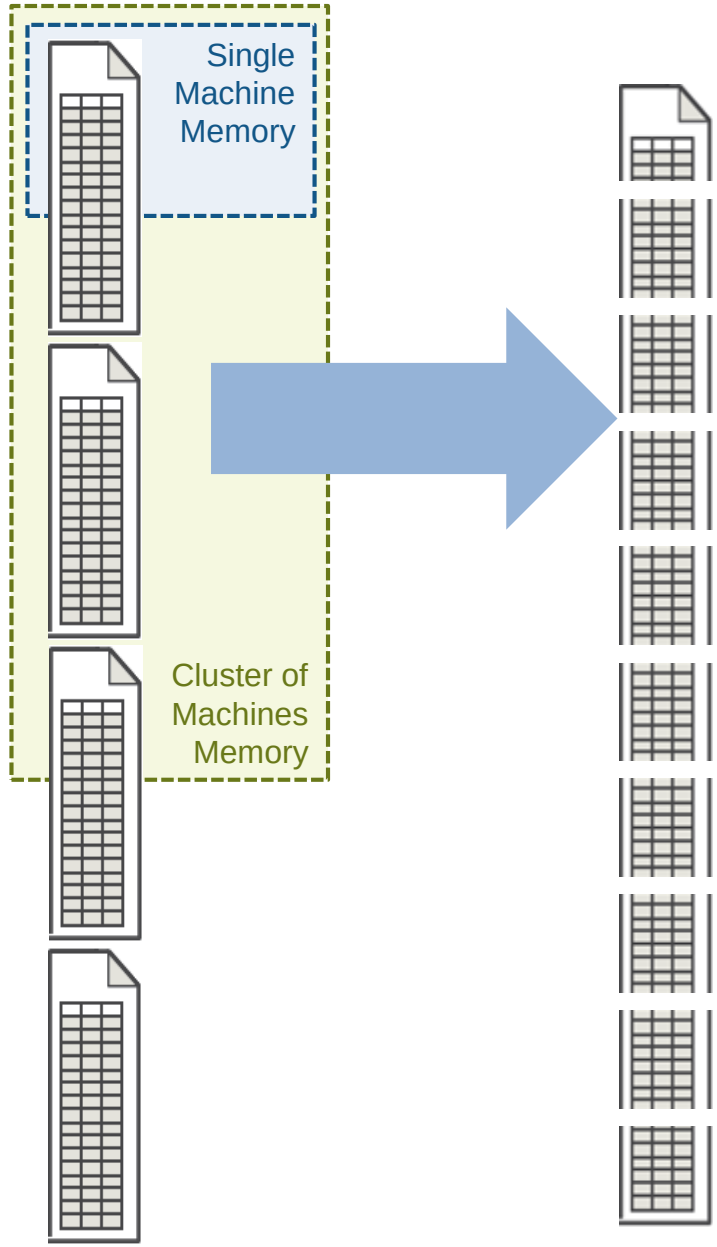
`tall` arrays R2016b

- Data is in one or more files
- Typically tabular data
- Files stacked vertically



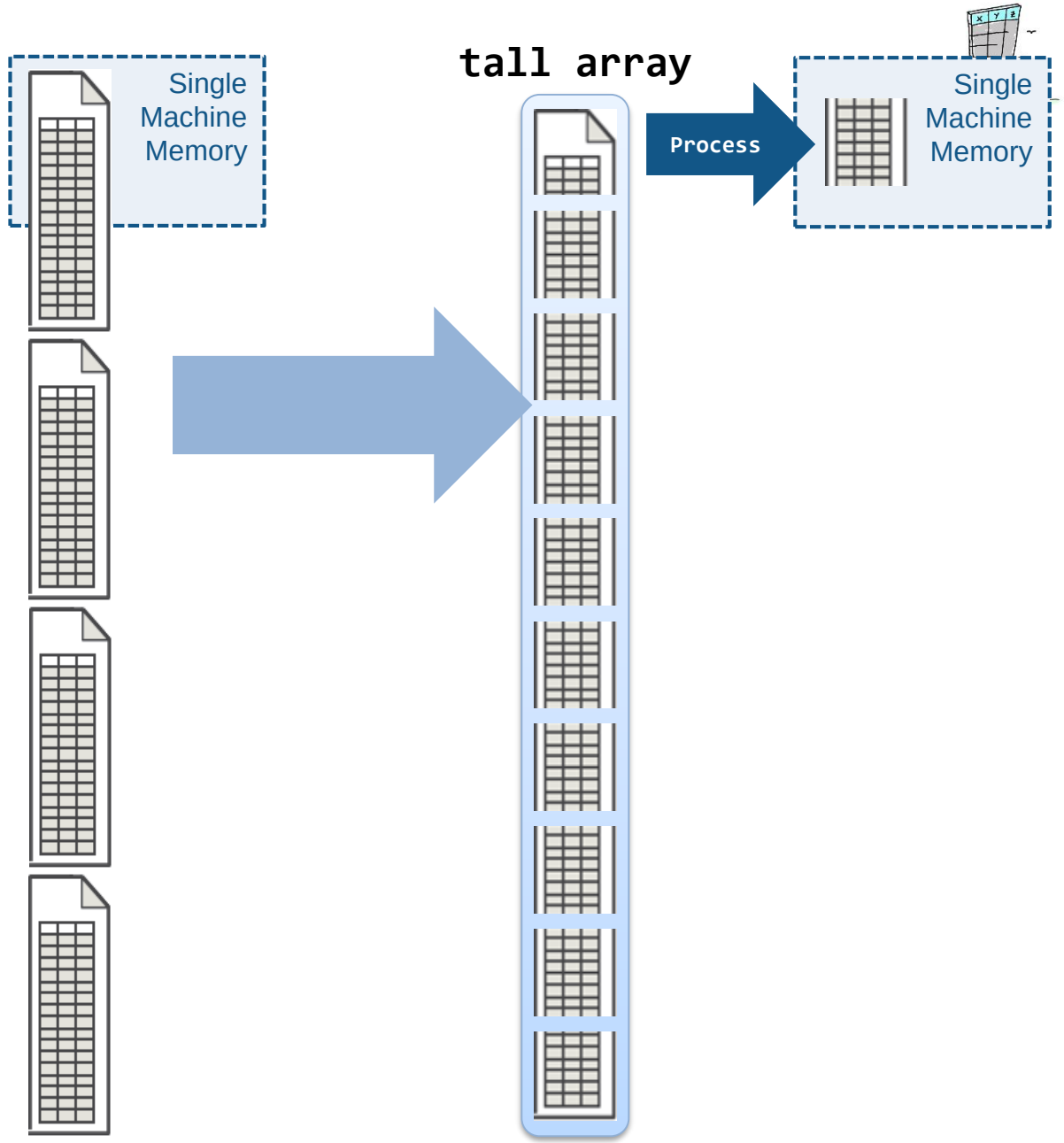
`tall` arrays R2016b

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



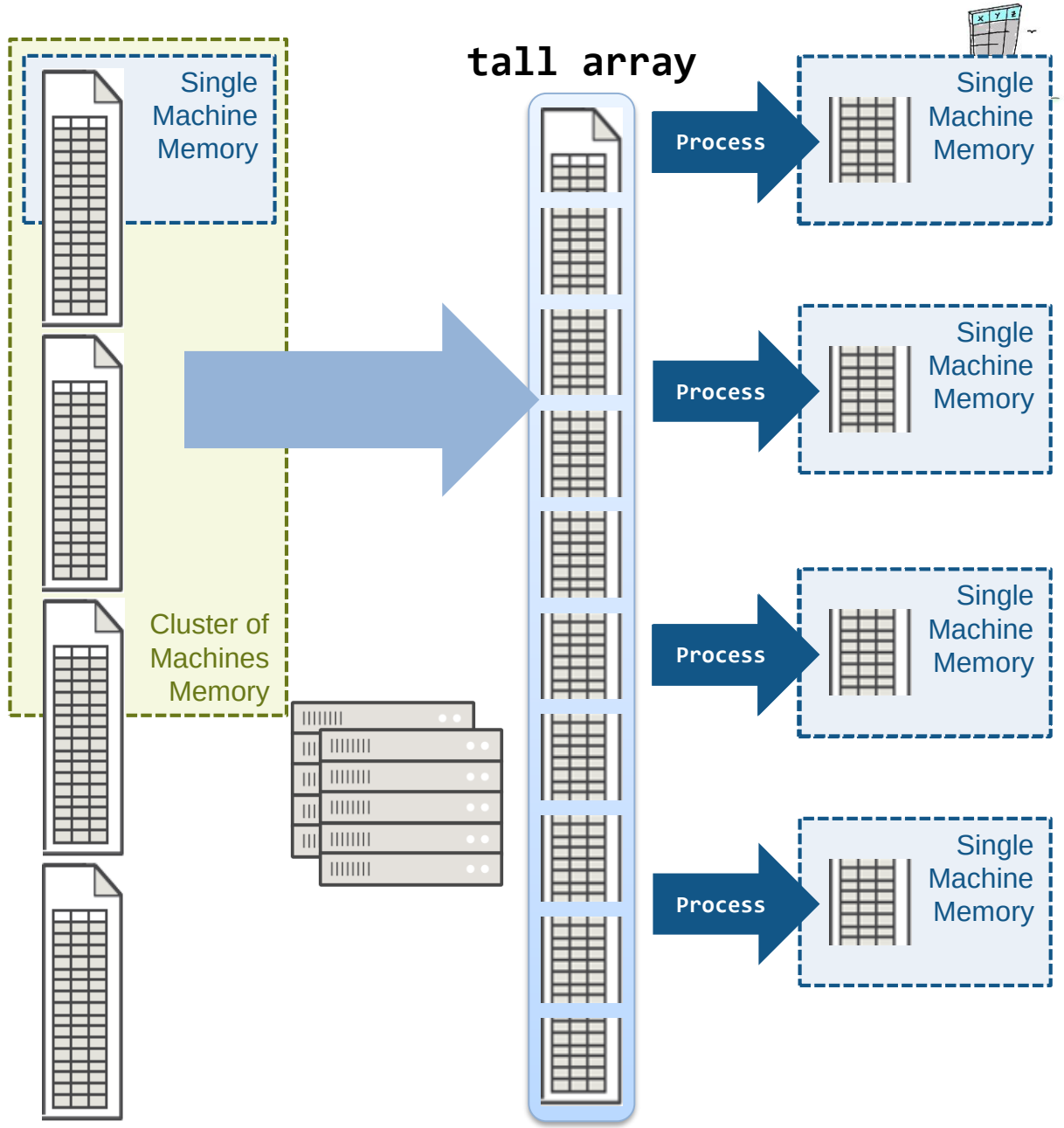
`tall` arrays R2016b

- 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 R2016b

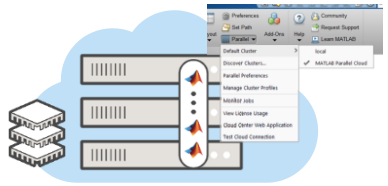
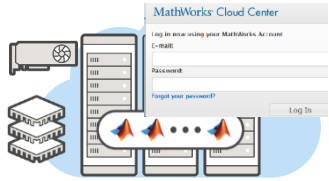
- Process several “chunks” at once with Parallel Computing Toolbox
- Scale up to clusters with MATLAB Distributed Computing Server

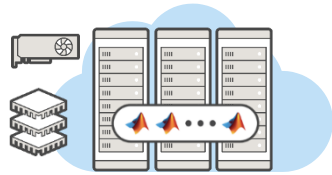


Agenda

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

Scale your applications beyond the desktop



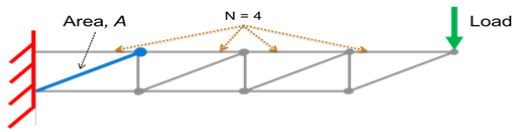
Option	Parallel Computing Toolbox	MATLAB Parallel Cloud	MATLAB Distributed Computing Server *for Amazon EC2*	MATLAB Distributed Computing Server *for Custom Cloud*	MATLAB Distributed Computing Server
Description	Explicit desktop scaling	Single-user, basic scaling to cloud	Scale to EC2 with some customization	Scale to custom cloud	Scale to clusters
Maximum workers	No limit	16	256	No limit	No limit
Hardware	Desktop	MathWorks Compute Cloud	Amazon EC2	Amazon EC2, Microsoft Azure,Others	Any
Availability	Worldwide	United States and Canada	United States, Canada and other select countries in Europe	Worldwide	Worldwide

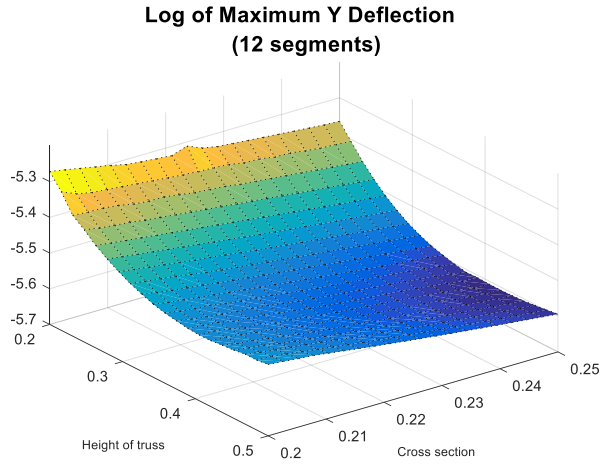
Learn More: Parallel Computing on the Cloud

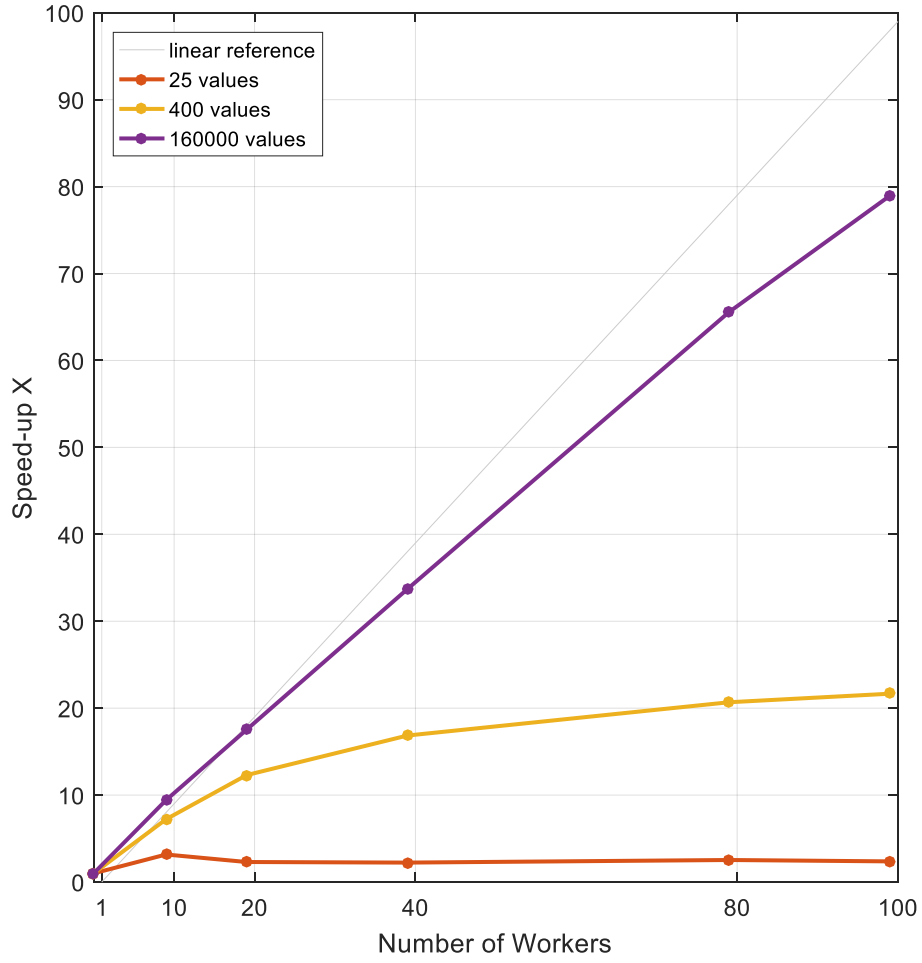
Why parallel computing matters

Scaling case study with a compute cluster

$$M\ddot{x} + C\dot{x} + Kx = F$$







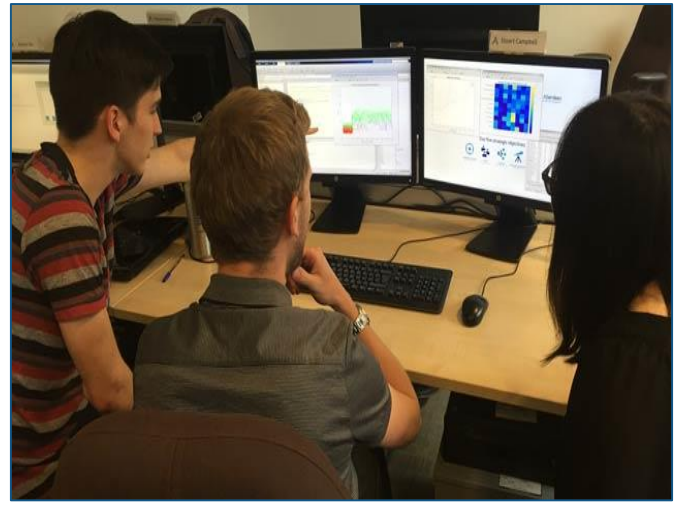
Workers in pool	Compute time (minutes)		
	160e3 values	400 values	25 values
1	140	0.38	0.03
10	15	0.05	0.01
20	8.0	0.03	0.01
40	4.2	0.02	0.01
80	2.1	0.02	0.01
100	1.8	0.02	0.01

Processor: Intel Xeon E5-class v2
16 physical cores per node
MATLAB R2016a

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

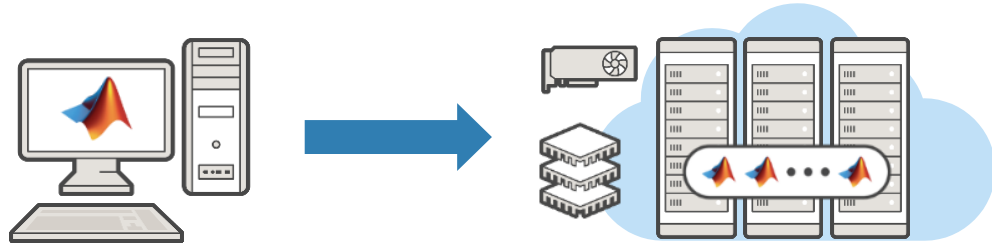
Why parallel computing:

- 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

What's new in 16b and 17a?



R2016b

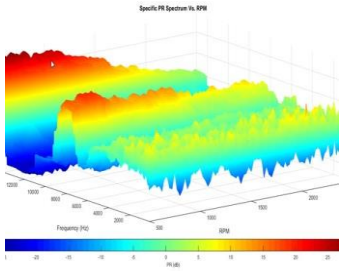
- `tall` array support for big data
- Measure data sent to workers using `ticBytes` and `tocBytes`
- Cloud offerings with K80-equipped GPUs

R2017a

- Simplified parallel Simulink simulations using `parsim`
- Send data to client using `DataQueue` and `PollableDataQueue`
- Train a single deep learning network with multiple CPUs or multiple GPUs

Summary and Takeaways

- Speed up your MATLAB and Simulink applications without being an expert
  - Reduce computation time by using multi core machines or GPUs
- Leverage the Parallel Computing Toolbox to reduce Computation Time: `parfor`, `gpuArray`, `parsim`
- 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



Automotive Test Data Analysis and Visualization
Validation time reduced by 40-50%
3-4 months of development time saved

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

Design and Build Wave Energy Farm
Sensitivity studies accelerated 12x

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

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

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.

Learn More

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

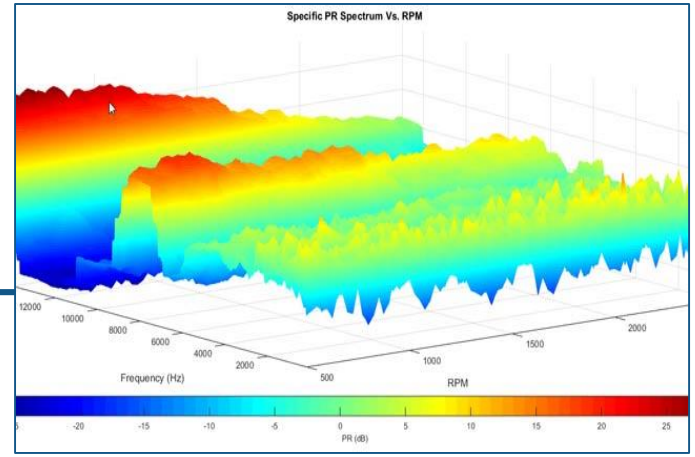
Why Parallel Computing

- 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

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



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

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

"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

Link to user story

Lockheed Martin Builds Discrete-Event Models to Predict F-35 Fleet Performance

F-35s ready for flight.

Challenge

Predict F-35 fleet performance to minimize life-cycle costs and maximize mission readiness

Solution

Build a discrete-event model of the fleet with Simulink and SimEvents, use MATLAB Distributed Computing Server to accelerate thousands of simulations, and interpolate the results with Neural Network Toolbox

Results

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

"By building a model with Simulink and SimEvents and running discrete-event simulations on a computer cluster, we rapidly identified many opportunities to maximize F-35 fleet performance while minimizing development and execution efforts."

Justin Beales
Lockheed Martin

Link to user story

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

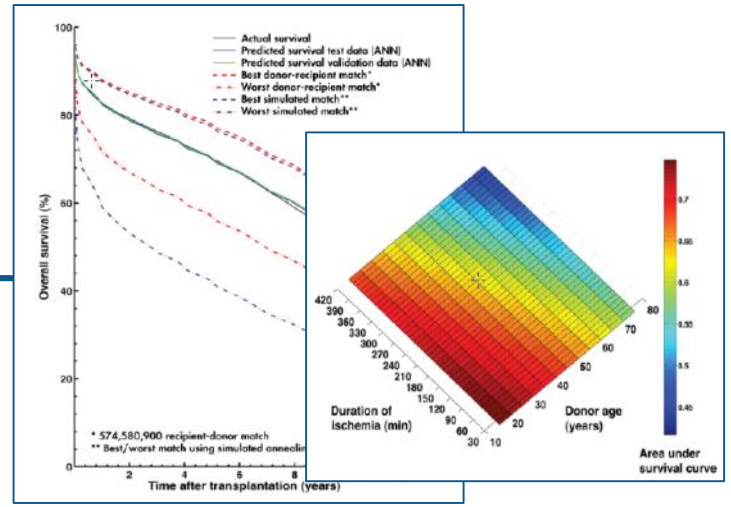
Improve long-term survival rates for heart transplant recipients by identifying optimal recipient and donor matches

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

Link to user story