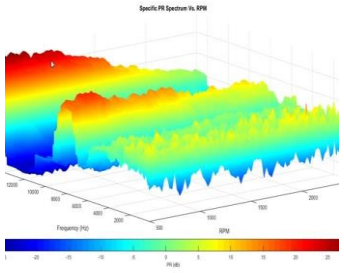
Parallel Computing in MATLAB

Sam Marshalik
Reece Teramoto

Practical application of parallel computing

- Why parallel computing?
  - Need faster insight to bring competitive products to market quickly
  - Computing infrastructure is broadly available (multicore desktops, GPUs, clusters)

- Why parallel computing with MATLAB
  - Leverage computational power of more hardware
  - Accelerate workflows with minimal to no code changes to your original code
  - Focus on your engineering and research, not the computation



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 Deep Learning Toolbox

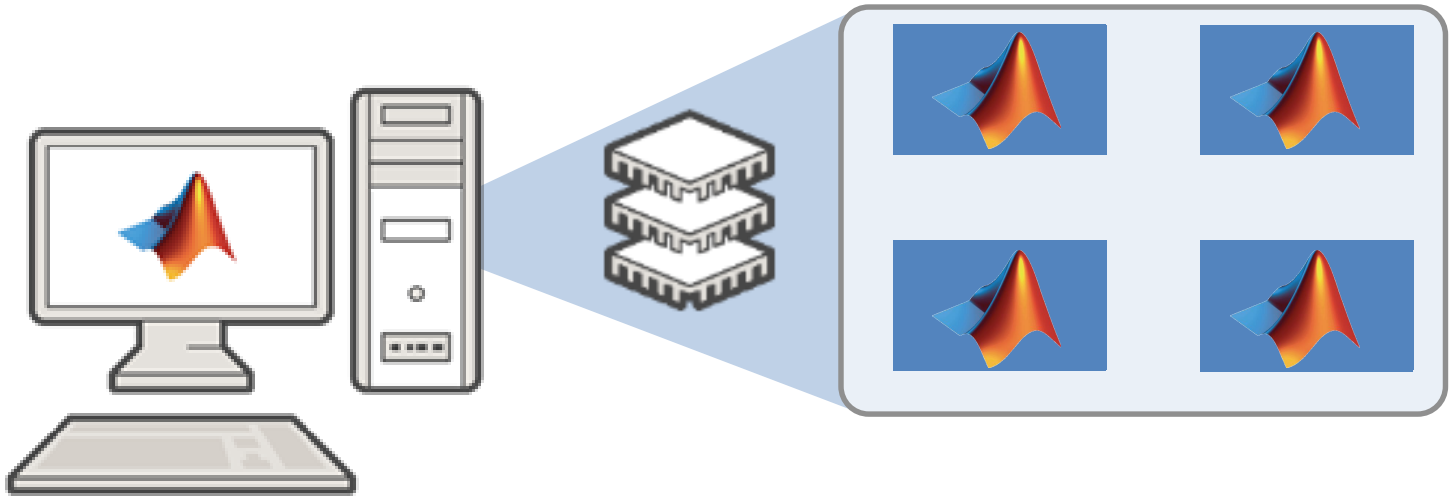
Calculating Derived Market Data
Implementation time reduced by months
Updates loaded 8X faster

Agenda

- Getting started with parallel computing in MATLAB
- Scaling beyond the desktop to clouds and clusters
- Big Data
- Accelerate applications with NVIDIA GPUs
- Summary

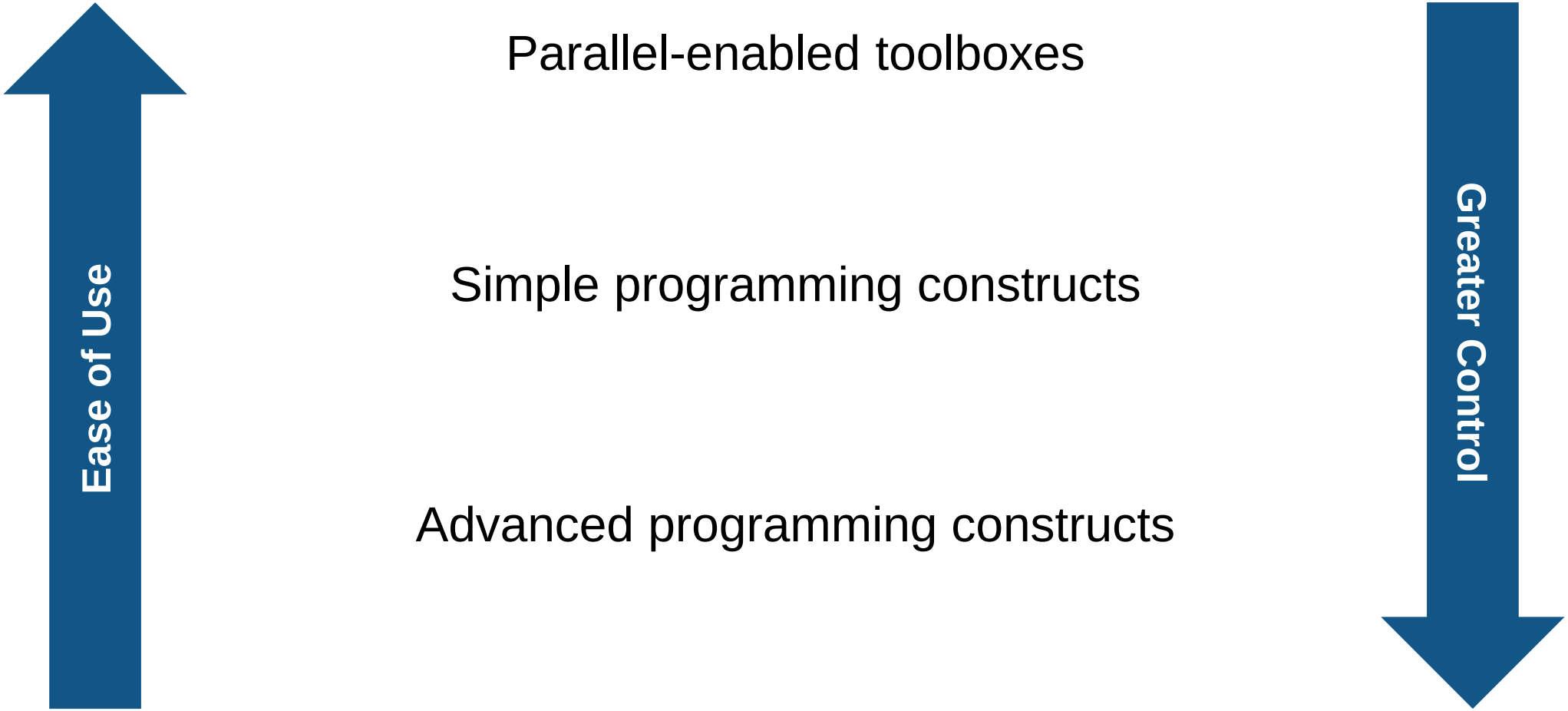
Parallel Computing Paradigm

Multicore Desktops



Parallel Computing Toolbox

Accelerating MATLAB and Simulink Applications



Parallel-enabled Toolboxes (MATLAB® Product Family)

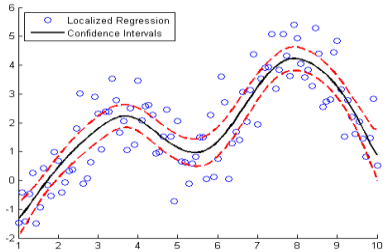
Enable acceleration by setting a flag or preference

Image Processing

Batch Image Processor, Block Processing, GPU-enabled functions

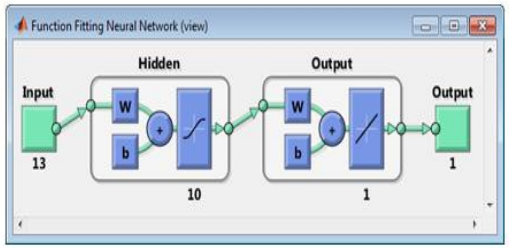
Statistics and Machine Learning

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



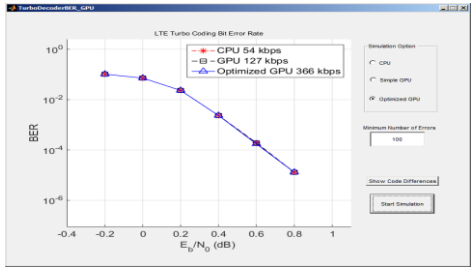
Deep Learning

Deep Learning, Neural Network training and simulation



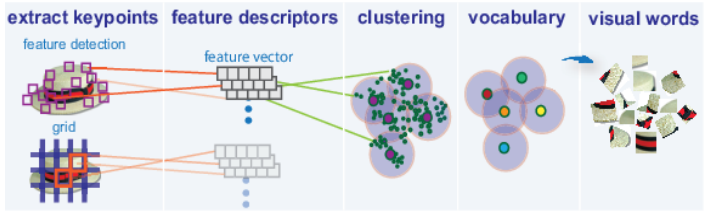
Signal Processing and Communications

GPU-enabled FFT filtering, cross correlation, BER simulations



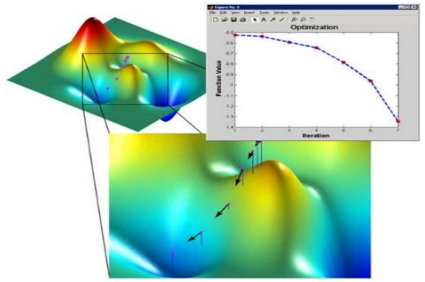
Computer Vision

Bag-of-words workflow



Optimization

Estimation of gradients



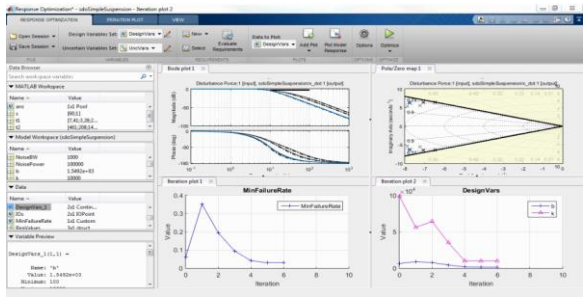
[Other Parallel-enabled Toolboxes]

Parallel-enabled Toolboxes (Simulink® Product Family)

Enable parallel computing support by setting a flag or preference

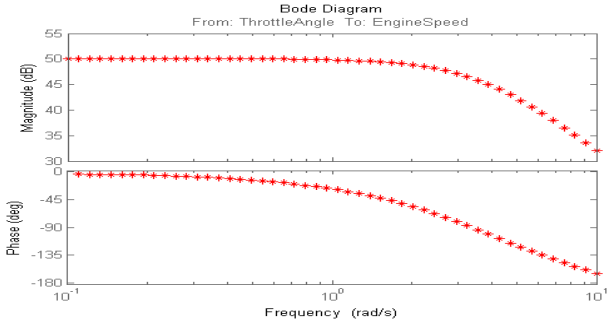
Simulink Design Optimization

Response optimization, sensitivity analysis, parameter estimation



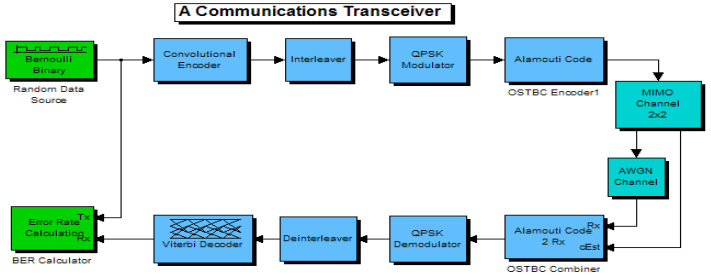
Simulink Control Design

Frequency response estimation



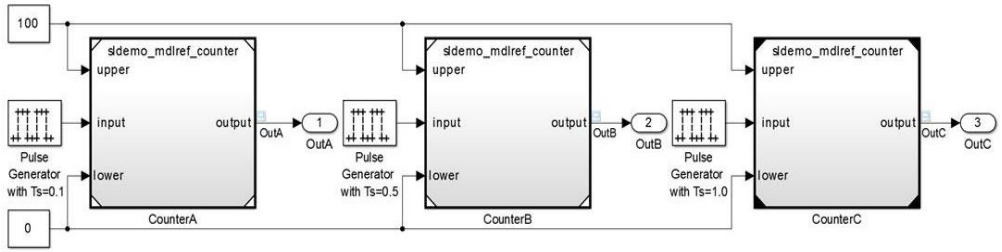
Communication Systems Toolbox

GPU-based System objects for Simulation Acceleration



Simulink/Embedded Coder

Generating and building code

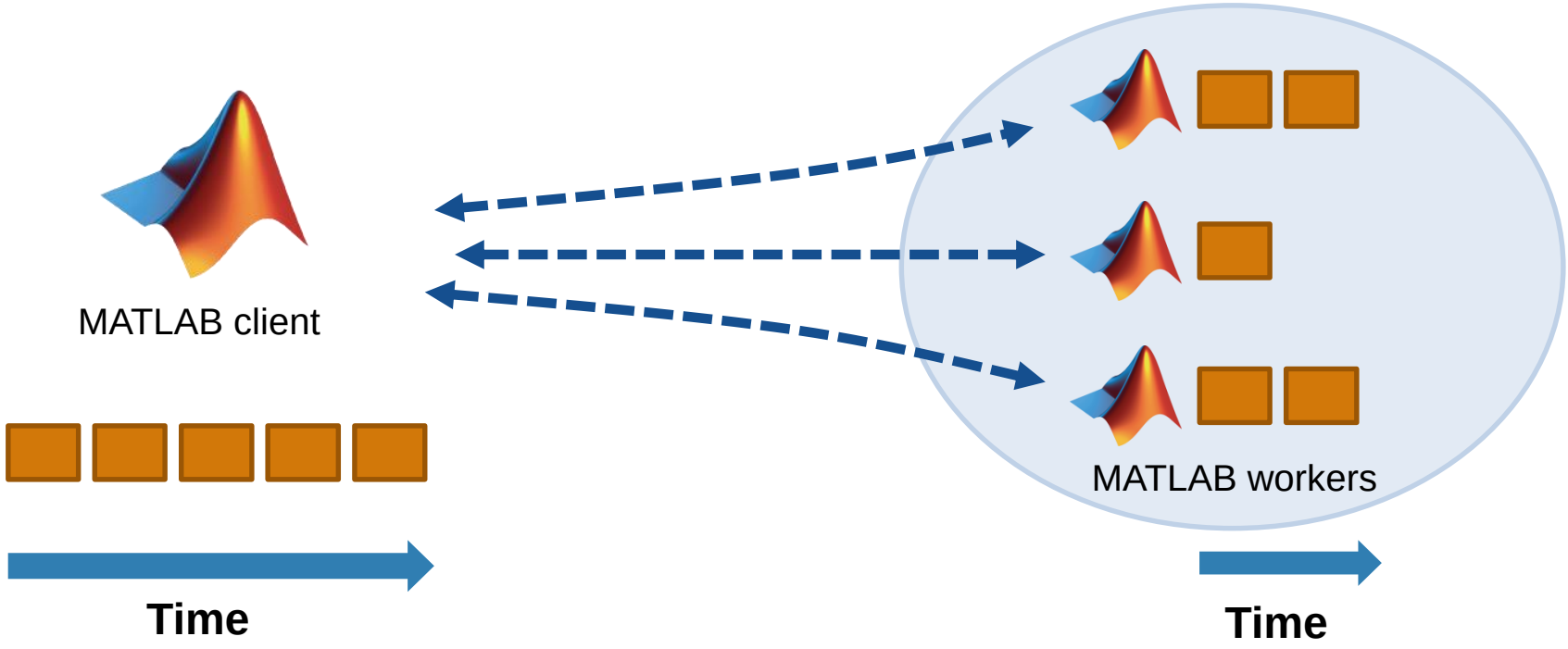


[Other Parallel-enabled Toolboxes]

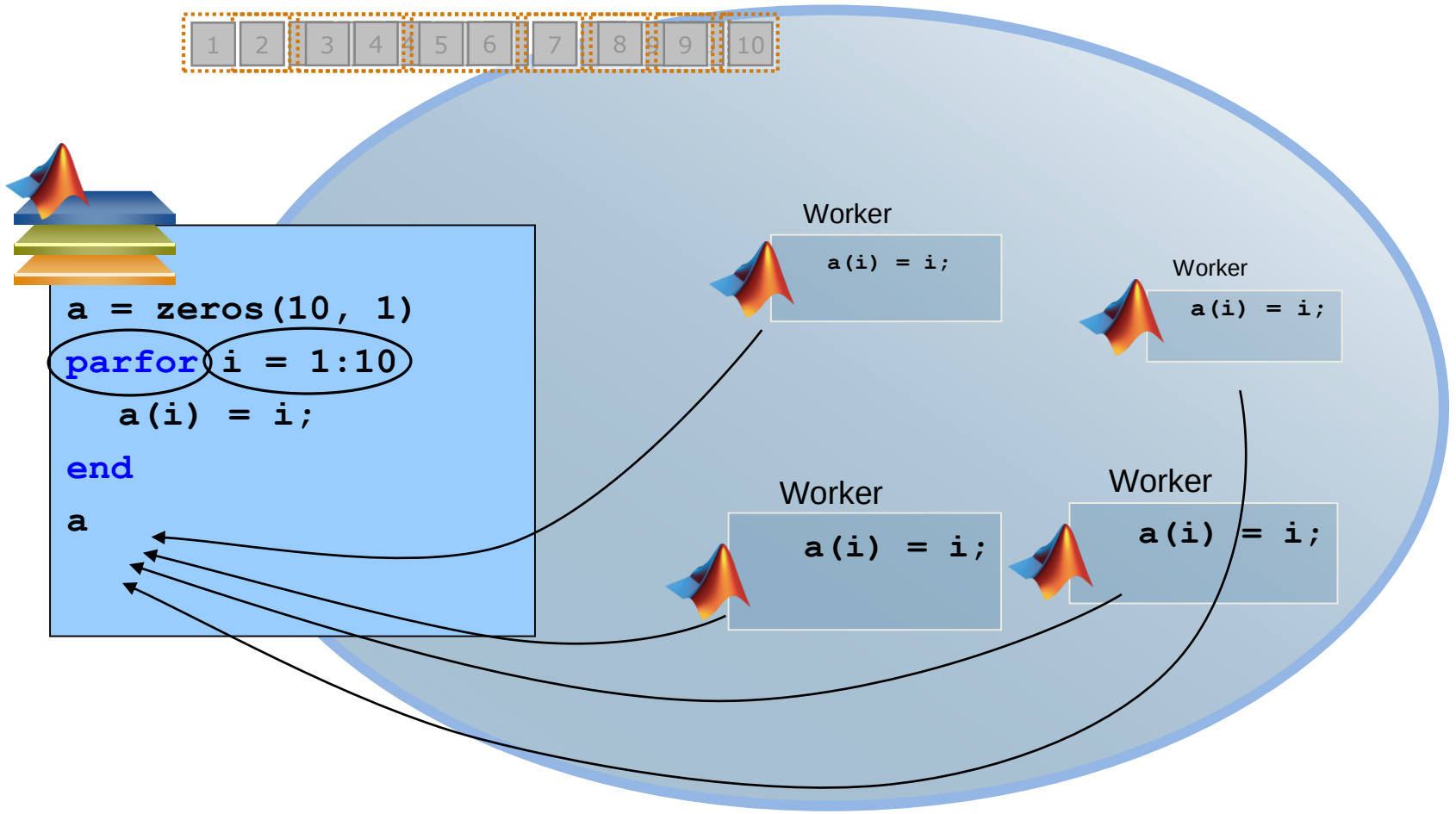
Explicit Parallelism: Independent Tasks or Iterations

Simple programming constructs: `parfor, parfeval`

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



Mechanics of `parfor` Loops



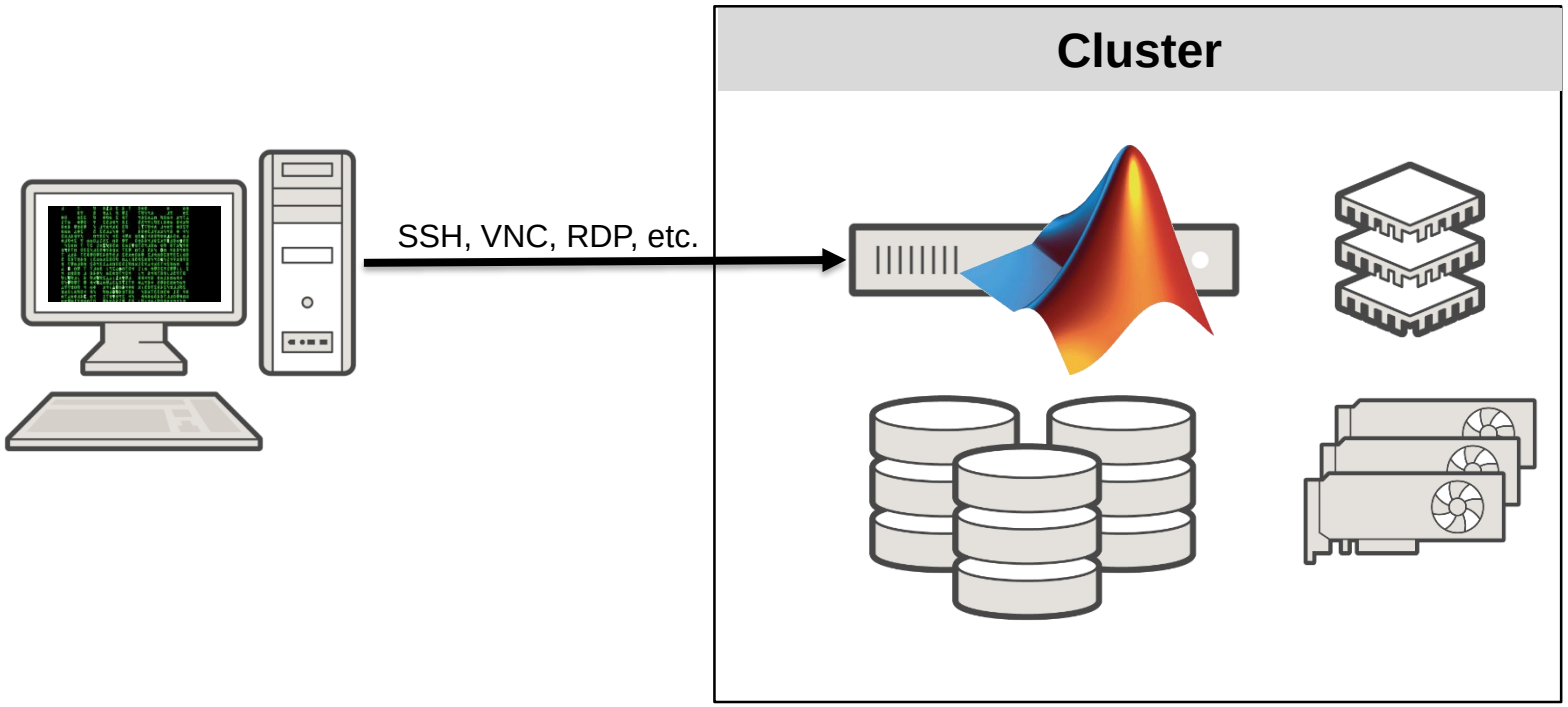
Key functionality

	Description	Functionality	Ease of use	Control
Parallel Enabled Toolboxes	Ready to use parallelized functions in MathWorks tools	• MATLAB and Simulink parallel enabled functions • Toolbox integration	Turnkey-automatic	Minimal (presets)
Common programming constructs	• Constructs that enable you to easily parallelize your code	• parfor • gpuArray • batch • distributed/tall • parsim • parfeval	Simple	Some
Advanced programming constructs	• Advanced parallelization techniques	• spmd • arrayfun/pagefun • CUDAKernel • MapReduce • MATLAB Spark API	Advanced	Extensive

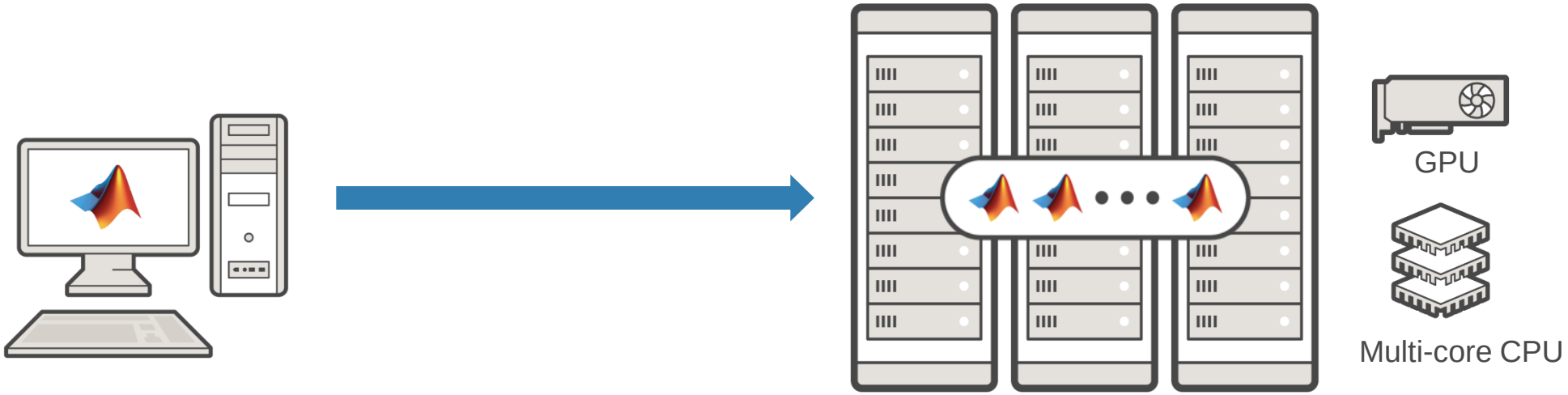
Agenda

- Getting started with parallel computing in MATLAB
- Scaling beyond the desktop to clouds and clusters
- Big Data
- Accelerate applications with NVIDIA GPUs
- Summary

Running MATLAB on a Cluster



Migrate Execution to a Cluster Environment with MATLAB Parallel Server



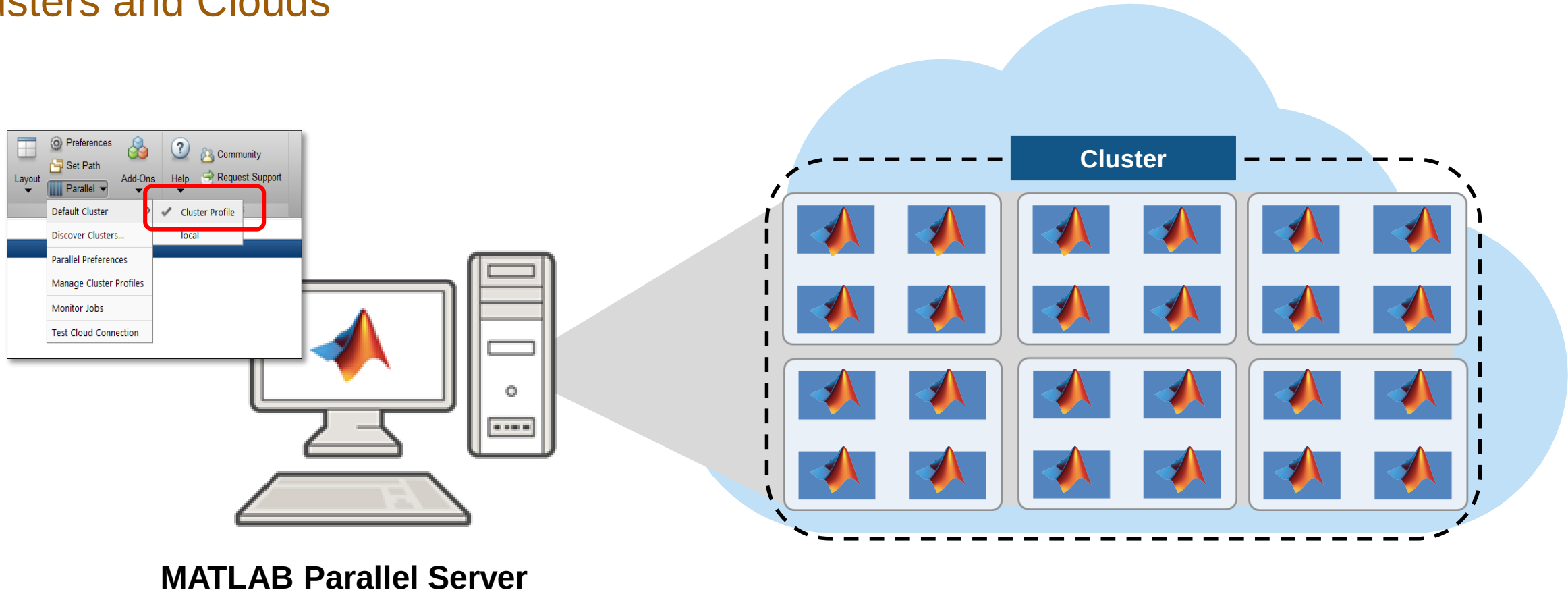
MATLAB

Parallel Computing Toolbox

MATLAB Parallel Server

Parallel Computing Paradigm

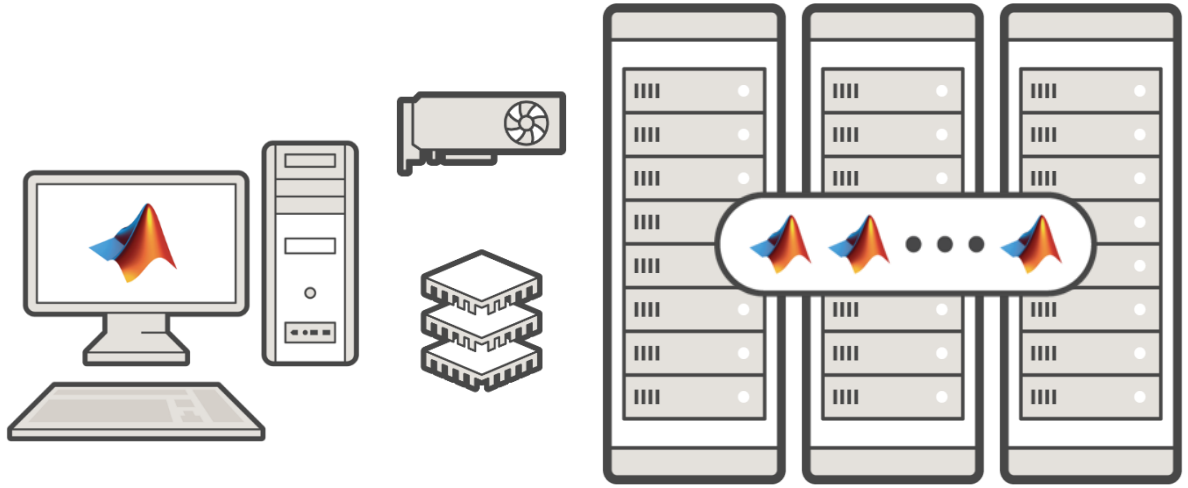
Clusters and Clouds



MATLAB Parallel Server

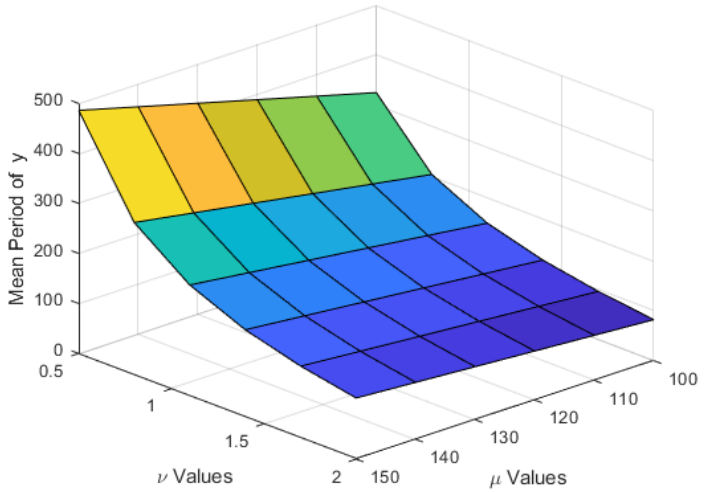
Take Advantage of Cluster Hardware

- Offload computation:
  - Free up desktop
  - Access better computers
- Scale speed-up:
  - Use more cores
  - Go from hours to minutes
- Scale memory:
  - Utilize distributed arrays
  - Solve larger problems without re-coding algorithms

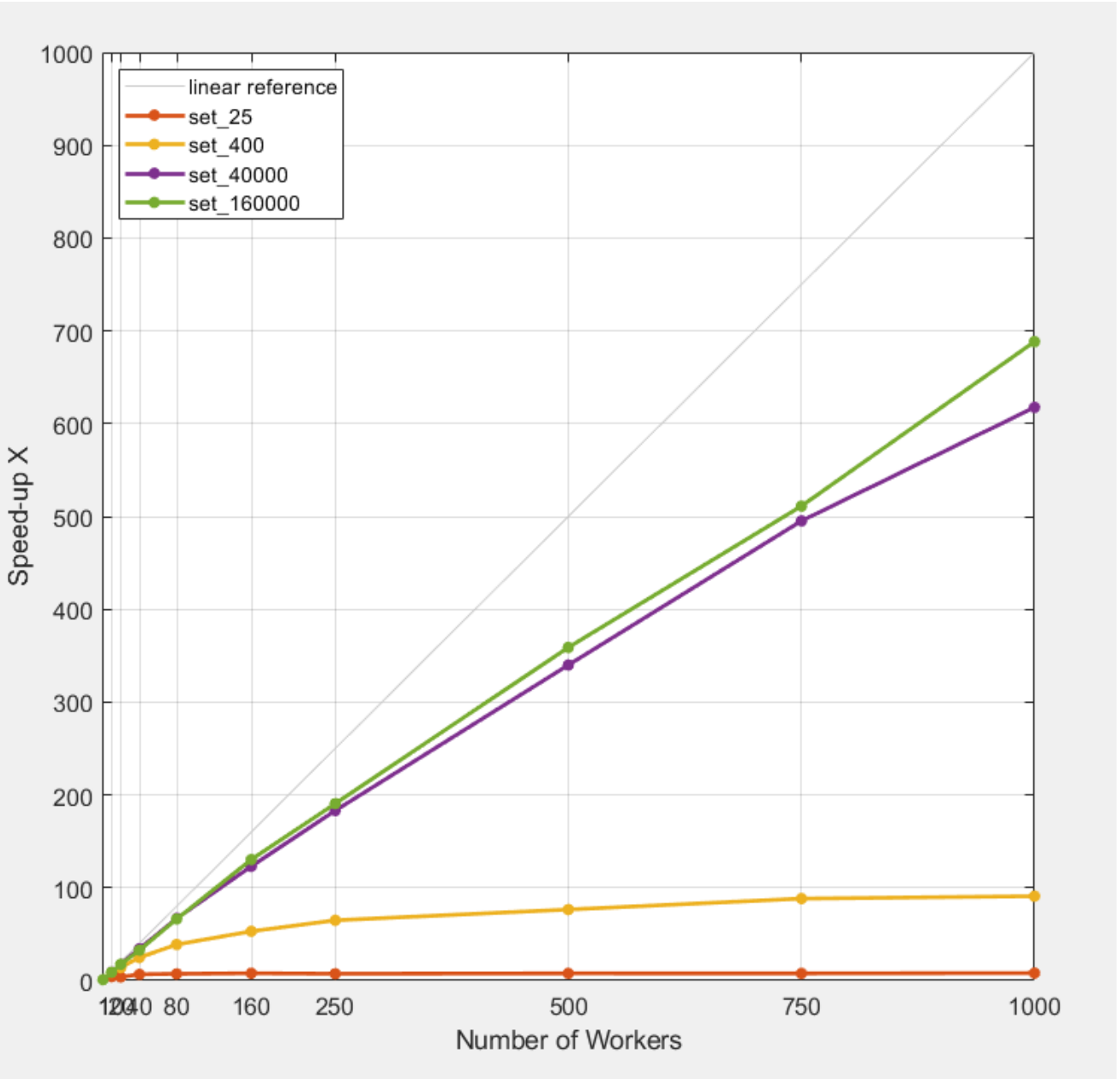


Why parallel computing matters

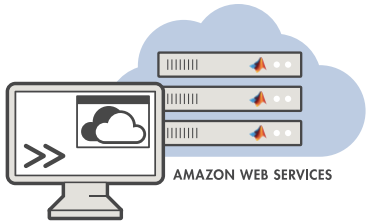
Scaling with a compute cluster

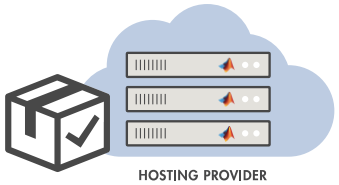


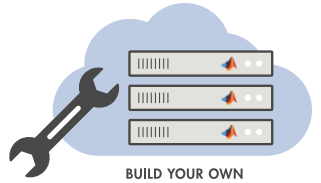
	Compute time (min)			
Workers in pool	160000 values	40000 values	400 values	25 values
1	1238.3	319.56	3.42	0.24
10	139.39	35.59	0.39	0.06
20	71.03	18.13	0.25	0.06
40	37.93	9.34	0.14	0.04
80	18.68	4.78	0.09	0.03
160	9.5	2.6	0.06	0.03
250	6.5	1.75	0.05	0.03
500	3.45	0.94	0.04	0.03
750	2.42	0.65	0.04	0.03
1000	1.8	0.52	0.04	0.03



Scale your applications on the desktop and beyond







Option	Parallel Computing Toolbox	MATLAB Parallel Server			
		managed by ***Cloud Center***	*managed by* ***Hosting Providers***	*for Custom Cloud* ***Reference Architecture***	***On Premise***
Description	Explicit desktop scaling	Preconfigured clusters in Amazon Web Services	Cloud solutions from MathWorks partners	Custom infrastructure for AWS, Azure, and others	Scale to clusters in your organization
Maximum workers	No limit	1024	Defined by Hosting Provider	No limit	No limit
Availability	Worldwide	Worldwide	Worldwide	Worldwide	Worldwide
License Options	• Network License Manager • LNU	• Online Licensing • Perpetual, Term *	• Network License Manager • Online Licensing • Perpetual, Term *	• Online Licensing • Perpetual, Term *	• Network License Manager • Online Licensing • Perpetual, Term *

https://www.mathworks.com/products/matlab-parallel-server/get-started.html *On-demand available in US and Europe

Agenda

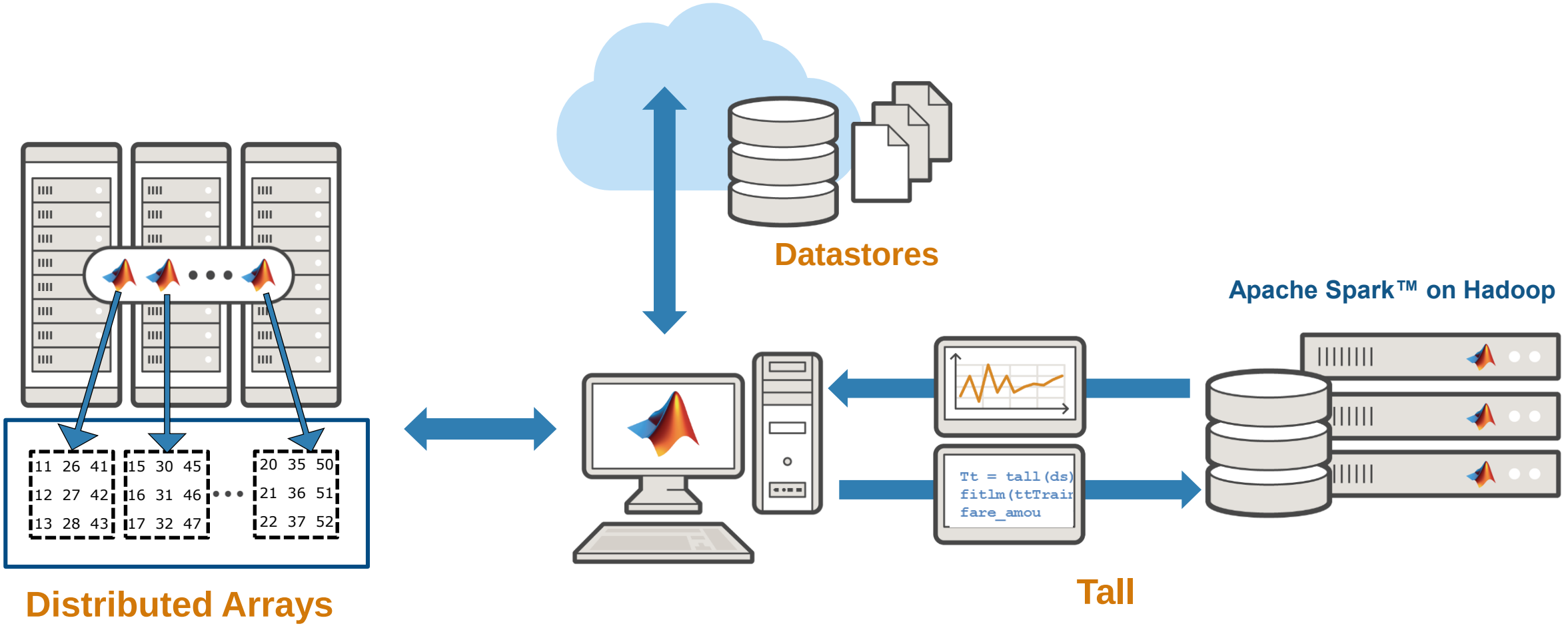
- Getting started with parallel computing in MATLAB
- Scaling beyond the desktop to clouds and clusters
- Big Data
- Accelerate applications with NVIDIA GPUs
- Summary

Big solutions

Wouldn’t it be nice if you could:

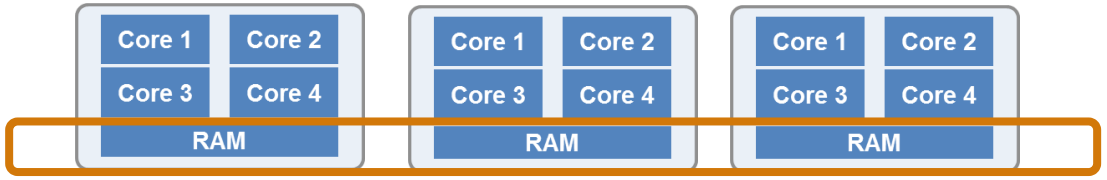
- Easily access data however it is stored
- Prototype algorithms quickly using small data sets
- Scale up to big data sets running on large clusters
- **Using the same intuitive MATLAB syntax you are used to**

Big Data capabilities in MATLAB



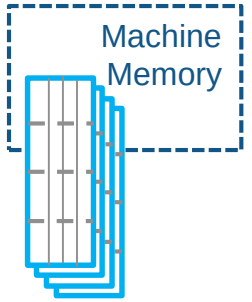
Tall and Distributed Data

- Distributed Data
  - Large matrices using the combined memory of a cluster



- Common Actions
  - Matrix Manipulation
  - Linear Algebra and Signal Processing

- Tall Data
  - Columnar data that does not fit in memory of a desktop or cluster



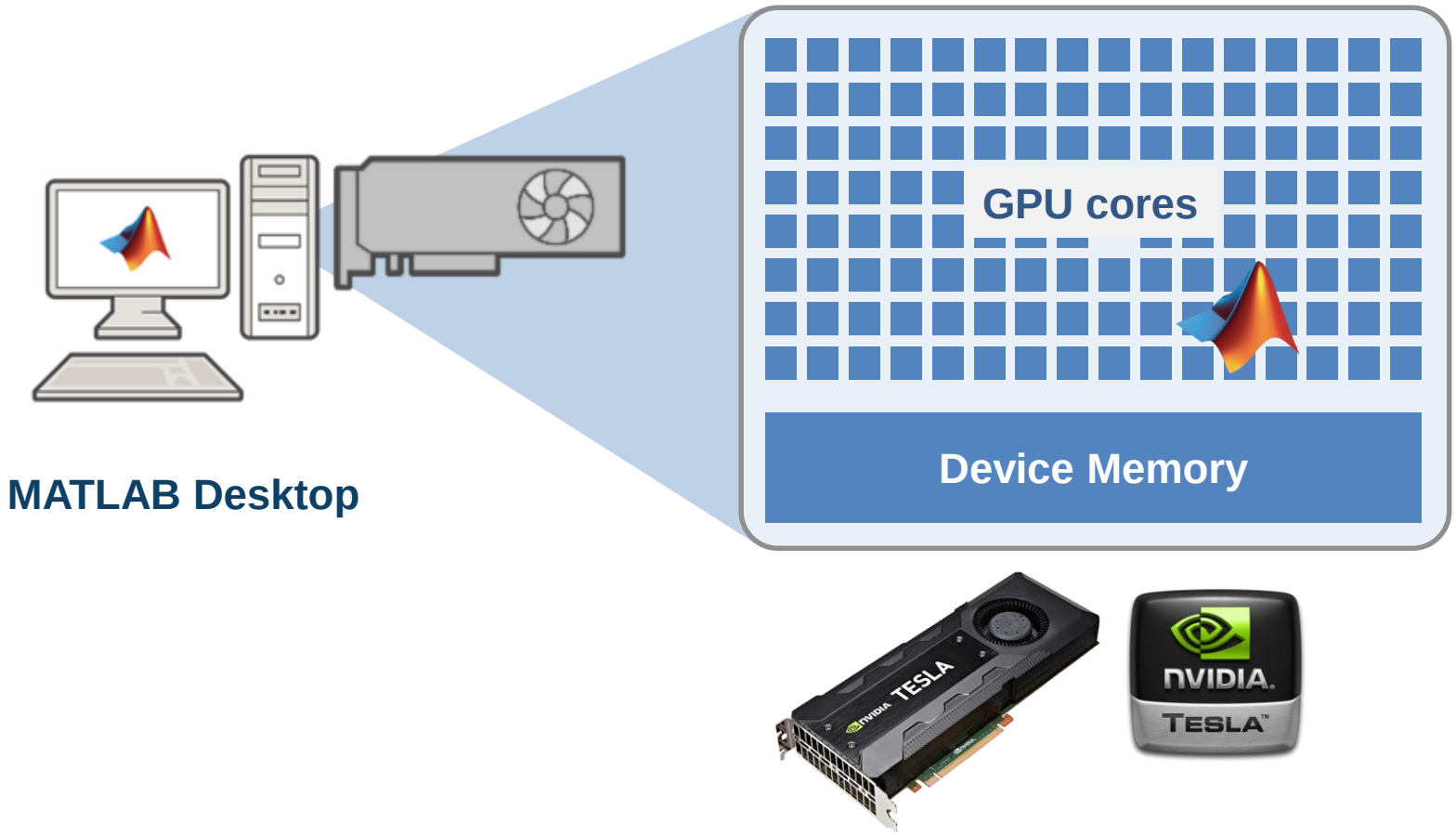
- Common Actions
  - Data manipulation, math, statistics
  - Summary visualizations
  - Machine learning

Agenda

- Getting started with parallel computing in MATLAB
- Scaling beyond the desktop to clouds and clusters
- Big Data
- Accelerate applications with NVIDIA GPUs
- Summary

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
- 500+ GPU-enabled MATLAB functions
- Additional GPU-enabled Toolboxes
  - Deep Learning
  - Image Processing and Computer Vision
  - Communications
  - Signal Processing
  - Statistics and Machine Learning

Transfer Data To GPU From Computer Memory

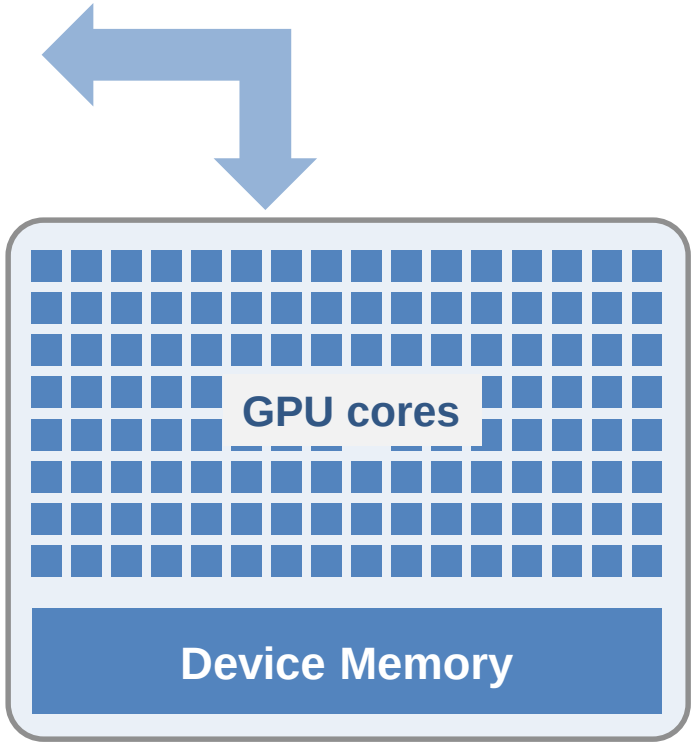
```
A=gpuArray(A);
```

Perform Calculation on GPU

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

Gather Data or Plot

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



Agenda

- Getting started with parallel computing in MATLAB
- Scaling beyond the desktop to clouds and clusters
- Big Data
- Accelerate applications with NVIDIA GPUs
- Summary

Summary

- Easily develop parallel MATLAB applications without being a parallel programming expert
- Speed up the execution of your MATLAB applications using additional hardware
- Develop parallel applications on your desktop and easily scale to a cluster when needed

Valuable Resources

- MATLAB Documentation
  - MATLAB → Advanced Software Development → Performance and Memory
  - Parallel Computing Toolbox
- Parallel and GPU Computing Tutorials
  - https://www.mathworks.com/videos/series/parallel-and-gpu-computing-tutorials-97719.html
- Parallel Computing with MATLAB
  - https://www.mathworks.com/solutions/parallel-computing.html

MATLAB Central Community

Every month, over **2 million** MATLAB & Simulink users visit MATLAB Central to get questions answered, download code and improve programming skills.

MATLAB Central

Learn | Contribute | Connect

MATLAB Answers: Q&A forum; most questions get answered in only **60 minutes**

File Exchange: Download code from a huge repository of free code including **tens of thousands** of open source community files

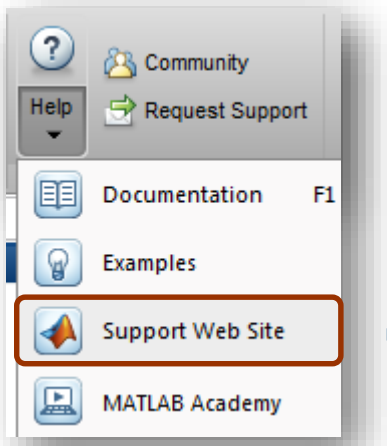
Cody: Sharpen programming skills while having fun

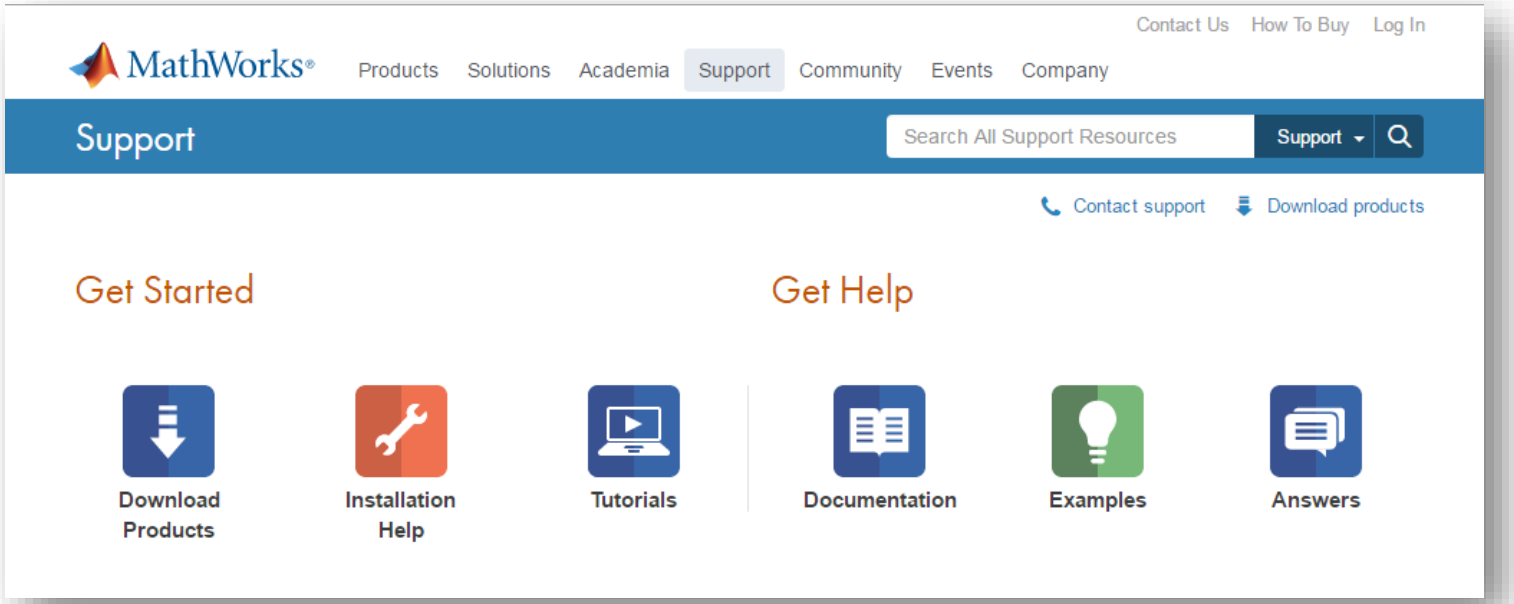
Blogs: Get the inside view from Engineers who build and support MATLAB & Simulink

ThingSpeak: Explore IoT Data

And more for you to explore…

Get Help



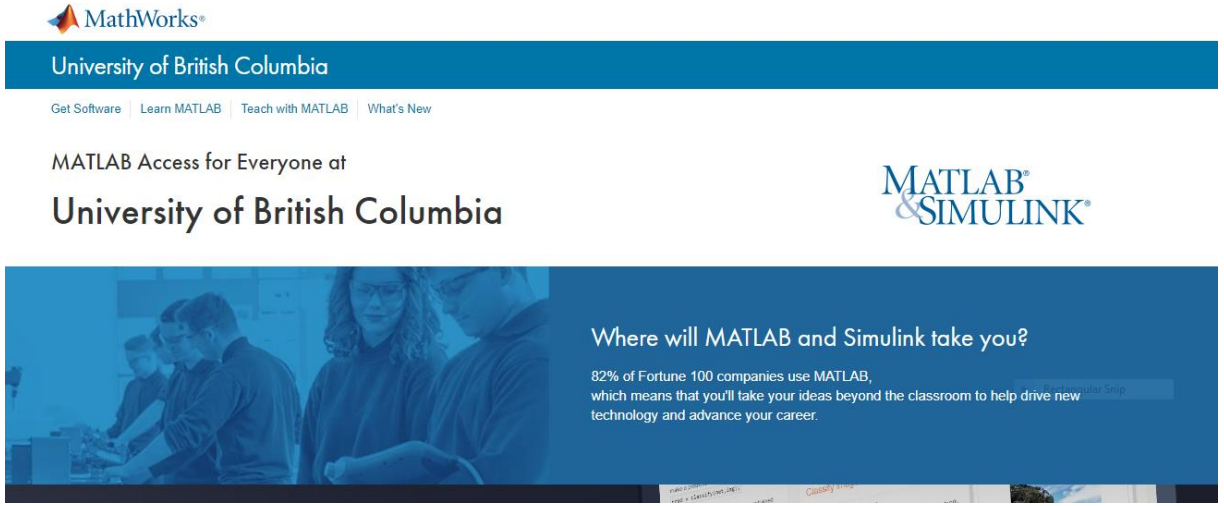


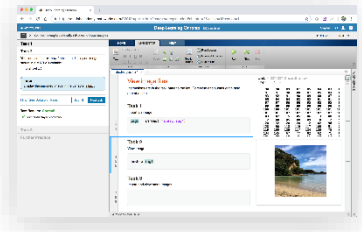
Quick access to:

- Self-serve tools
- Community knowledge base
- Support engineers

Campus-Wide License Overview

University & lab computers



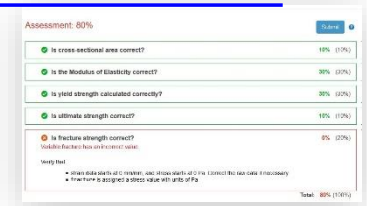


Self-paced online learning

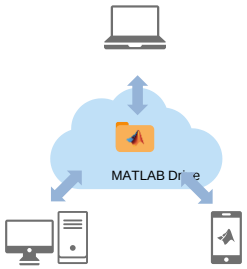
Online access

https://www.mathworks.com/academia/tah-portal/university-of-british-columbia-924490.html

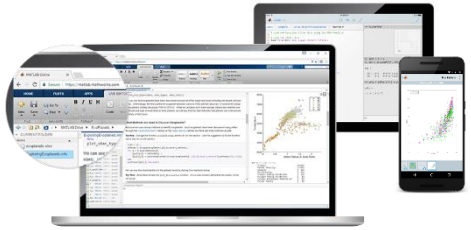
- License covers all faculty, staff, students and their devices
- Access on campus, in lab and field, and at home, including off-network
- Annual license
- Immediate tool availability for end users via self-serve portal
- Lower IT administration overhead
- Storage of existing perpetual licenses



Auto-graded homework

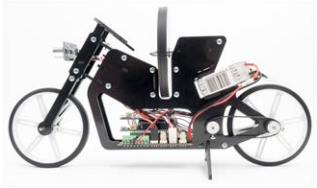


Cloud Storage & Sharing



Personal Computers & Mobile Devices

Clusters & HPC



Low-cost hardware support

MATLAB Access on Compute Canada Hardware

- Who can run MATLAB on Compute Canada hardware?
  - Any affiliated researcher, professor, student, or staff associated with a Canadian university
- What products are available?
  - MATLAB, Toolboxes, and the MATLAB Compiler Runtime
  - Versions ranging from R2014a to R2018b
- Where can I run MATLAB on Compute Canada hardware?
  - Cedar and Beluga offer open access
    - Available to anyone associated with a Canadian university
  - Graham
    - Bring your own license (BYOL) – use your existing licenses to gain access
    - Wiki explains how to ensure your institution is enabling BYOL

Running MATLAB on Compute Canada Clusters

https://docs.computecanada.ca/wiki/MATLAB

https://docs.computecanada.ca/wiki/Cedar
https://docs.computecanada.ca/wiki/Graham
https://docs.computecanada.ca/wiki/Béluga

Issues? Questions?

support@computecanada.ca

Connect to a Compute Canada Cluster

```
$ ssh smarshal@cedar.computecanada.ca
smarshal@cedar.computecanada.ca's password:
================================================================
Welcome to Cedar!

Change to job submission
------------------------
Effective immediately we will no longer allow jobs to be run in the /home
filesystem. This is to reduce the load on that file system and improve the
responsiveness for interactive work. If the command
readlink -f $(pwd) | cut -d/ -f2
(type exactly like that) returns "home", you are not permitted to submit jobs
from that directory. Transfer the files from that directory either to your
/project or /scratch directories and submit the job from there.

For information see: https://docs.computecanada.ca/wiki/Cedar
Email support@computecanada.ca for assistance and/or to report problems.
================================================================

[smarshal@cedar1 ~]$ _
```

Request a Node to Run MATLAB

```
[smarshal@cedar1 ~]$ # Go to $SCRATCH to request Slurm resources
[smarshal@cedar1 ~]$ cd $SCRATCH
[smarshal@cedar1 smarshal]$
[smarshal@cedar1 smarshal]$ # Request a single node with 8 CPU cores for 3 hours with 4Gb/core
[smarshal@cedar1 smarshal]$ salloc -n 1 --cpus-per-task=8 --mem-per-cpu=4Gb -t 3:00:00
salloc: Granted job allocation 20806110
salloc: Waiting for resource configuration
salloc: Nodes cdr544 are ready for job
[smarshal@cdr544 smarshal]$ _
```

Run MATLAB - Interactively

```
[smarshal@cdr1264 smarshal]$ # Load MATLAB onto the system path
[smarshal@cdr1264 smarshal]$ module load matlab
[smarshal@cdr1264 smarshal]$
[smarshal@cdr1264 smarshal]$ # Run MATLAB
[smarshal@cdr1264 smarshal]$ matlab -nodesktop
MATLAB is selecting SOFTWARE OPENGL rendering.
Opening log file:  /home/smarshal/java.log.43027

                                    < M A T L A B (R) >
                          Copyright 1984-2018 The MathWorks, Inc.
                           R2018b (9.5.0.944444) 64-bit (glnxa64)
                                      August 28, 2018


To get started, type doc.
For product information, visit www.mathworks.com.

>>
```

Run MATLAB - Interactively

```
>> % Query for the number of computationl threads
>> maxNumCompThreads

ans =

     8

>>
>> % Override default pool size with the number of available threads
>> p = parpool(8);
Starting parallel pool (parpool) using the 'local' profile ...
connected to 8 workers.
>>
>> % Run a parfor-loop with 48 iterations, pause for 2 seconds each
>> tic, parfor idx = 1:48, pause(2), end, toc
Elapsed time is 14.222245 seconds.
>>
>> % Close the pool
>> p.delete
Parallel pool using the 'local' profile is shutting down.
>> _
```

Grab the MATLAB batch scripts

```
[cedar5:~] # Go to your SCRATCH directory
[cedar5:~] cd $SCRATCH
[cedar5:/scratch/rsnorris]
[cedar5:/scratch/rsnorris] # Copy the workshop dir back to your SCRATCH
[cedar5:/scratch/rsnorris] cp -r /scratch/smarshal/workshop .
[cedar5:/scratch/rsnorris]
[cedar5:/scratch/rsnorris] # Check the workshop directory content
[cedar5:/scratch/rsnorris] cd workshop/
[cedar5:/scratch/rsnorris/workshop] ls
matlab.jobscript  my_parallel_app.m
[cedar5:/scratch/rsnorris/workshop] _
```

Run MATLAB – Batch

matlab.jobscript

```
#!/bin/bash

##SBATCH --account=def-<NAME>                    # Specify account to use
#SBATCH --cpus-per-task=8
#SBATCH --job-name=matlab
#SBATCH --mail-type=ALL
##SBATCH --mail-user=<EMAIL-ADDRESS>             # Receive e-mail notifications
#SBATCH --mem-per-cpu=4GB                       # REQUIRED: Memory per CPU core
#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --time=00:30:00                         # REQUIRED: Job runtime
##SBATCH --reservation=reservation-name         # If using a reservation

# ================================================================

# Load MATLAB onto the system path
# Should grab the latest MATLAB
# Load a specific MATLAB version by running:
# module load matlab/R2018b
module load matlab

# Run MATLAB
matlab -nodisplay -r my_parallel_app
```

Run MATLAB – Batch

my_parallel_app

```
function my_parallel_app

% Open a pool of workers
numWorkers = 8;
p = parpool(numWorkers);

% PARFOR
t0 = tic;
parfor idx = 1:(numWorkers*6)
    pause(2)
end

t = toc(t0)

% SPMD
spmd
   o = gop(@plus,labindex);
end

o{end}

% Compare our results to a cumulative summation
o = cumsum(1:numWorkers);
o(end)

% Close the pool
p.delete
```

Submit MATLAB Job to the Scheduler

```
[smarshal@cedar1 workshop]$ ls
matlab.jobscript  my_parallel_app.m
[smarshal@cedar1 workshop]$
[smarshal@cedar1 workshop]$ # Submit job to scheduler
[smarshal@cedar1 workshop]$ sbatch matlab.jobscript
Submitted batch job 20381766
[smarshal@cedar1 workshop]$
[smarshal@cedar1 workshop]$ # View job in the queue
[smarshal@cedar1 workshop]$ squeue -u $USER
   JOBID     USER      ACCOUNT           NAME  ST START_TIME        TIME_LEFT NODES CPUS       GRES MIN_MEM NODELIST (REASON)
20381766 smarshal def-razoumov         matlab   R 2019-05-02T07:47      29:50     1    8     (null)      4G cdr686 (None)
[smarshal@cedar1 workshop]$ _
```

View the Results

```
[smarshal@cedar1 workshop]$ ls
matlab.jobscript  my_parallel_app.m  slurm-20381766.out
[smarshal@cedar1 workshop]$
[smarshal@cedar1 workshop]$ # View the results
[smarshal@cedar1 workshop]$ cat slurm-20381766.out
Opening log file:  /home/smarshal/java.log.28233

                            < M A T L A B (R) >
                  Copyright 1984-2018 The MathWorks, Inc.
                   R2018b (9.5.0.944444) 64-bit (glnxa64)
                              August 28, 2018


To get started, type doc.
For product information, visit www.mathworks.com.

Starting parallel pool (parpool) using the 'local' profile ...
connected to 8 workers.

t =

   14.4726


ans =

    36


ans =

    36

Parallel pool using the 'local' profile is shutting down.
[smarshal@cedar1 workshop]$ _
```