

Large Data in MATLAB: A Seismic Data Processing Case Study

Stuart Kozola

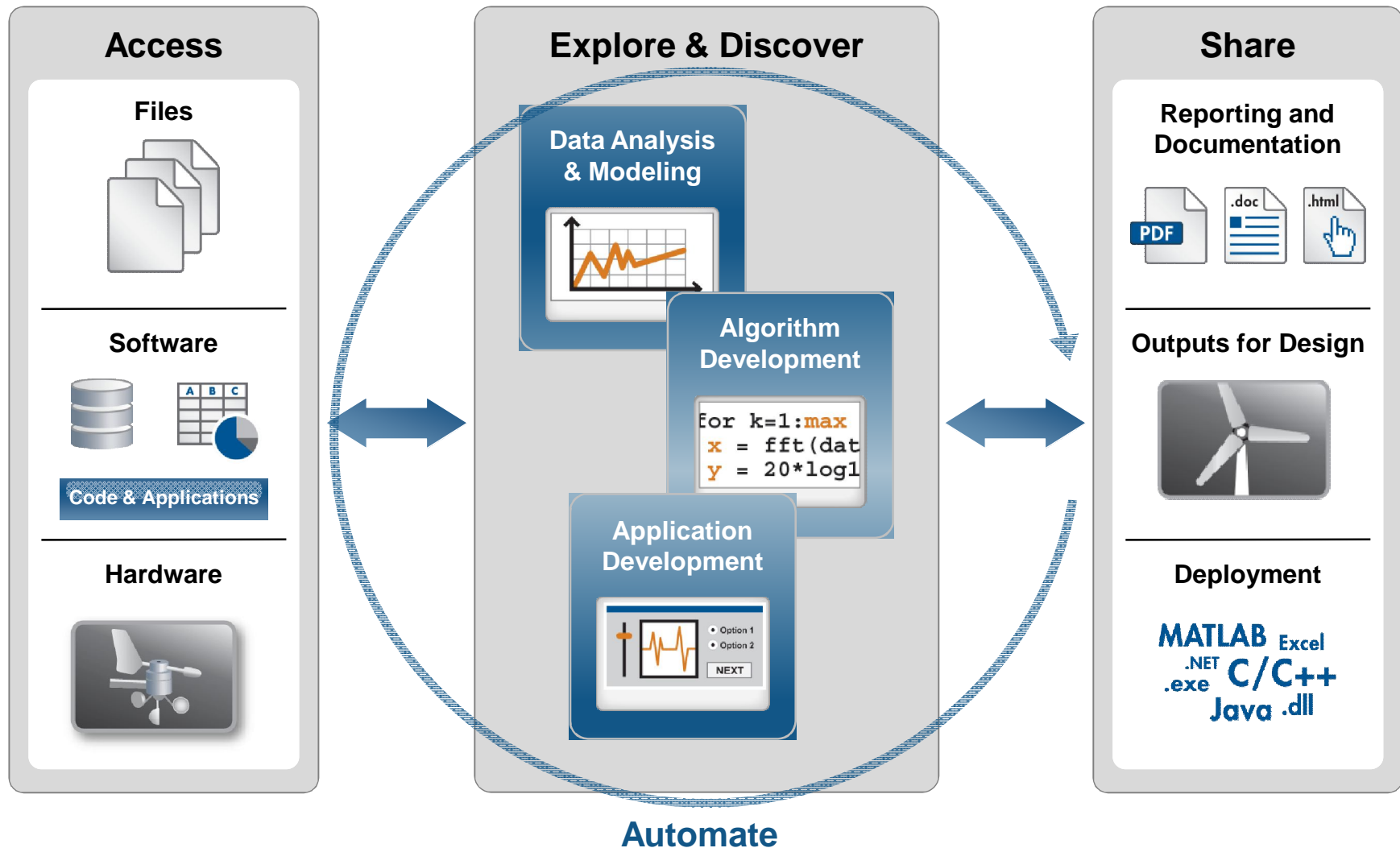
Agenda

- Challenges in working with large data
- Scaling up seismic analysis for large data
- MathWorks large data solutions
- Question and answer

Challenges in working with large data

- More data than available memory
 - From multiple files
 - From large files
 - From databases
 - Generated during analysis/simulations
- My “Toolbox” works on arrays that are “in memory” only
- Multiple versions of code required for
 - In-memory vs. virtual arrays
 - Parallel computations
 - GPUs

Data Analysis Tasks



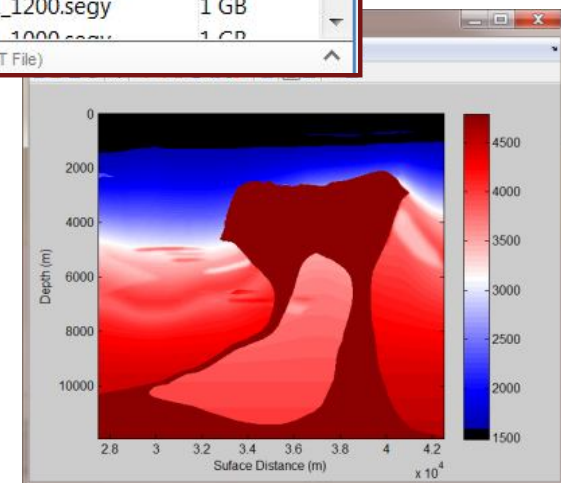
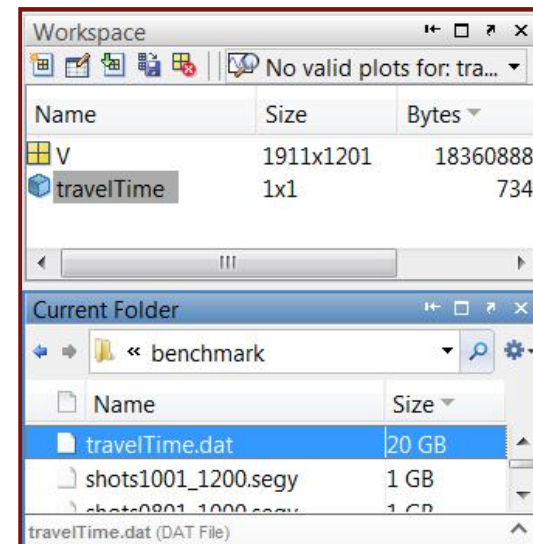
Demo: Scaling Up Seismic Analysis

Challenge:

- Developing a seismic analysis algorithm that can scale up to large data

Solution:

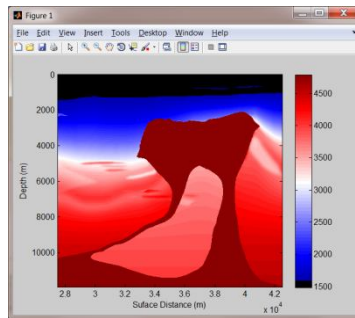
- Analyze seismic data files larger than available system memory
- Use parallel computing or GPUs for faster processing



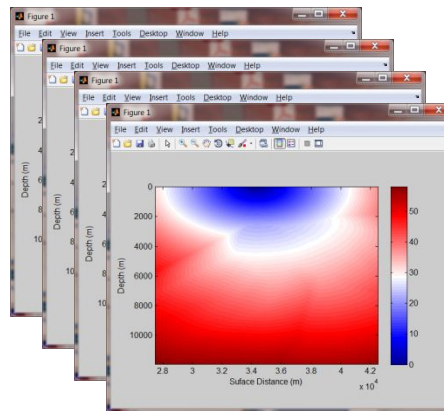
How do I scale my algorithm for large data?

Kirchhoff Migration

Velocity Model

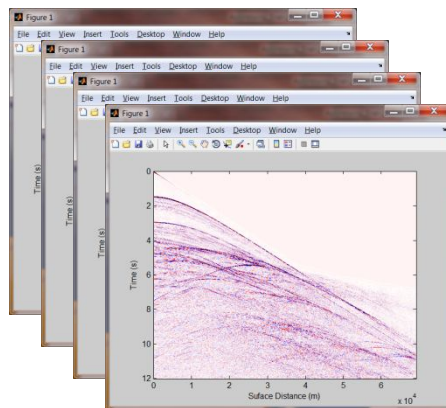


Travel Time Field

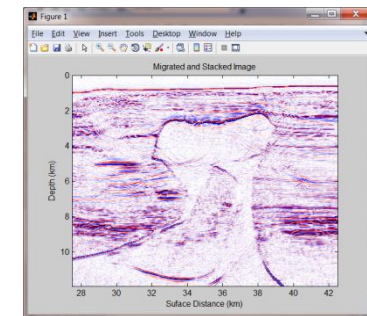


20 GB

Migrate(Shot, TravelTime)



8 GB

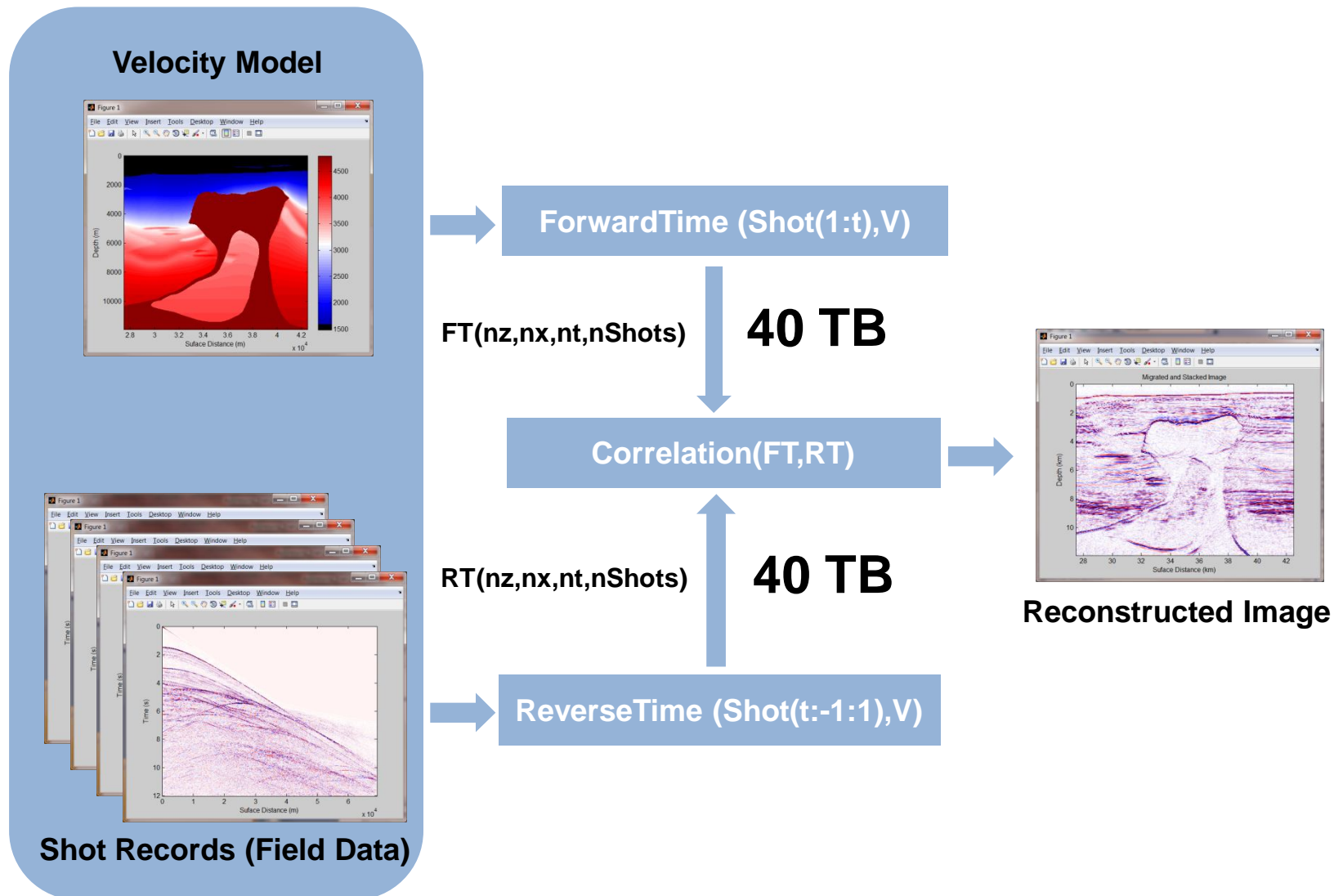


Reconstructed Image

Shot Records (Field Data)

How do I scale my algorithm for large data?

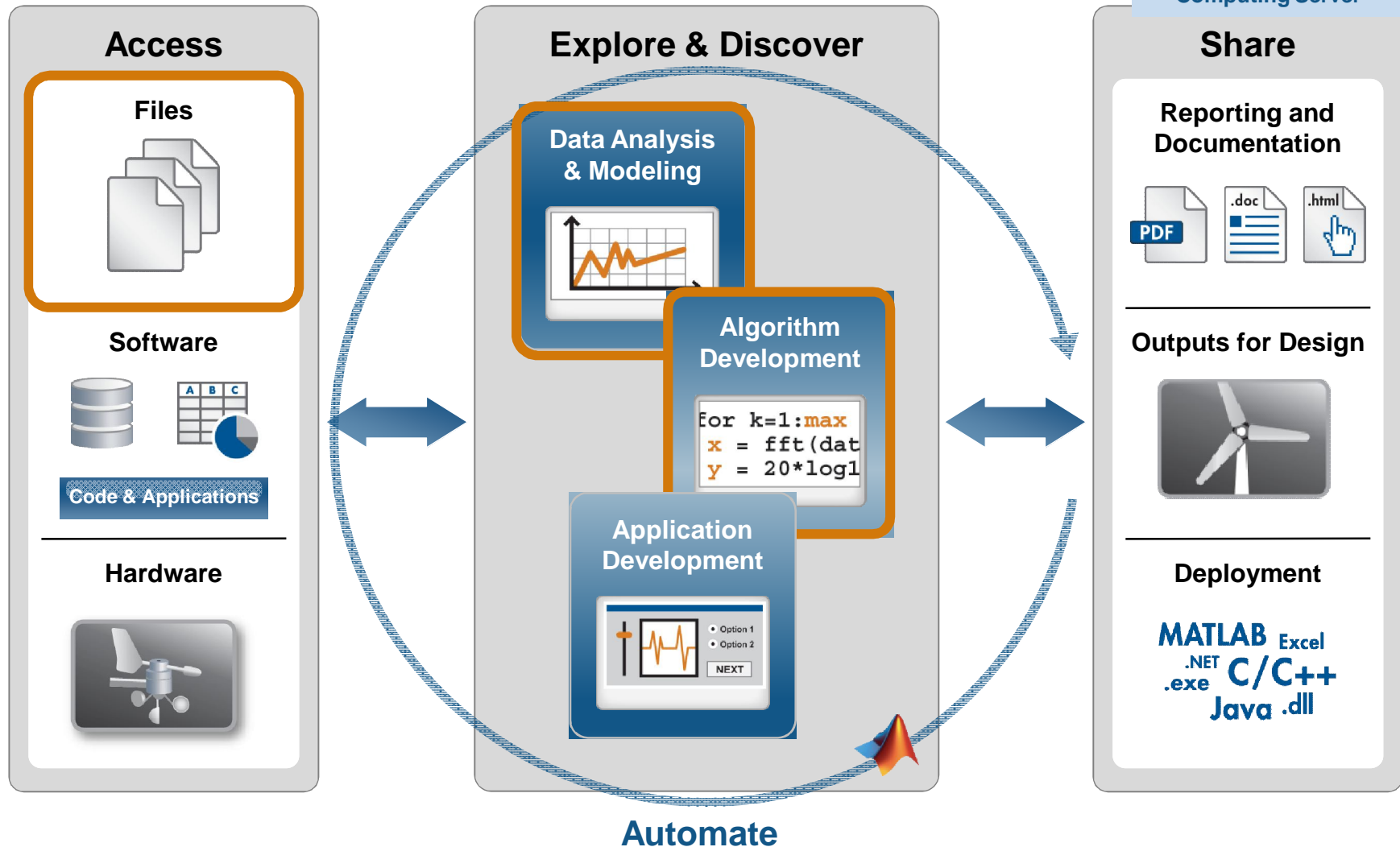
Reverse Time Migration



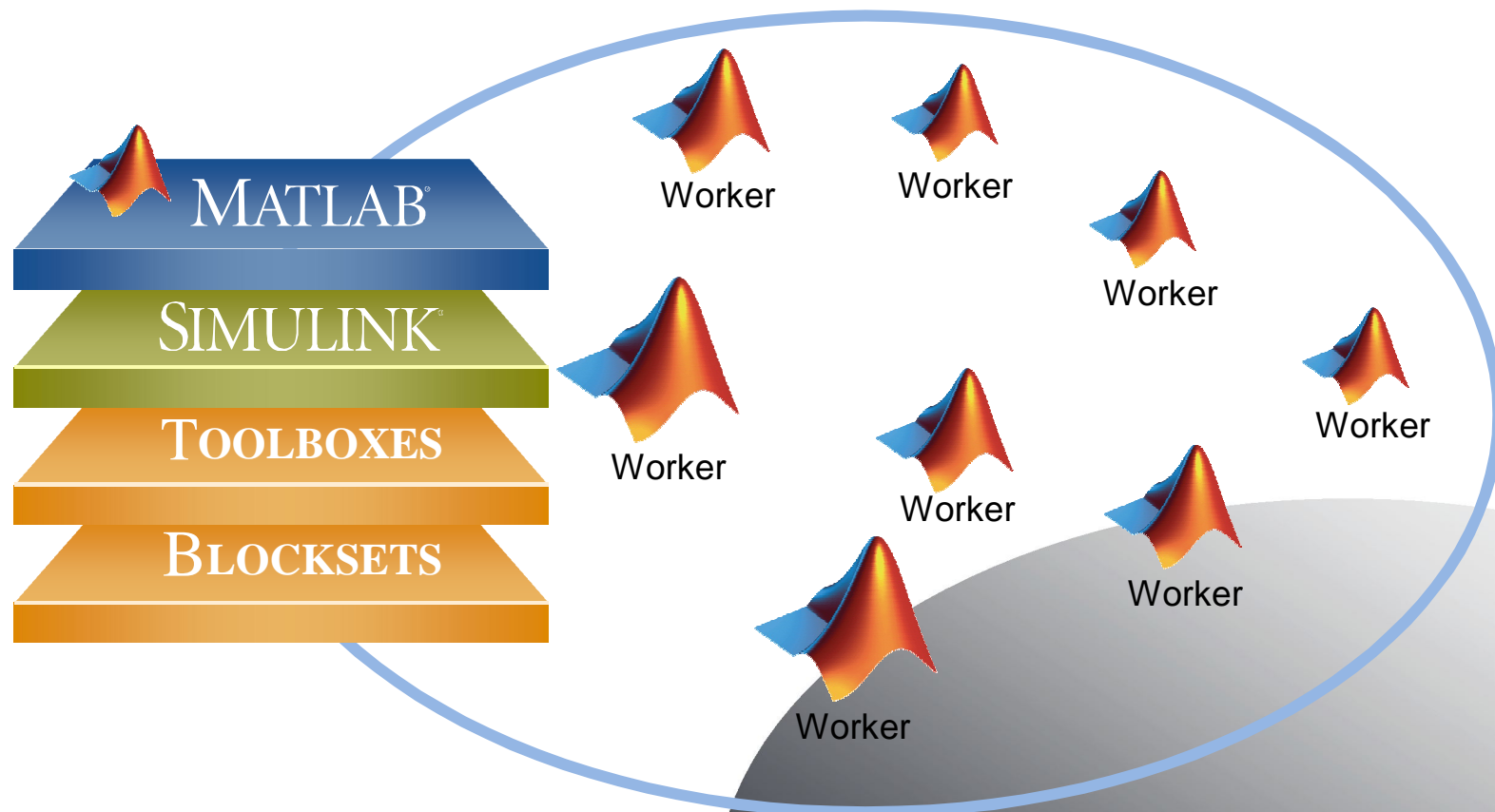
Demo: Scaling Up Seismic Analysis

Products Used

- MATLAB
- Parallel Computing Toolbox
- MATLAB Distributed Computing Server



Scaling MATLAB Applications



Parallel Computing Tools Address...

Task-Parallel

Long computations

- Multiple independent iterations

```
parfor i = 1 : n
    % do something with i
end
```

- Series of tasks

Task 1

Task 2

Task 3

Task 4

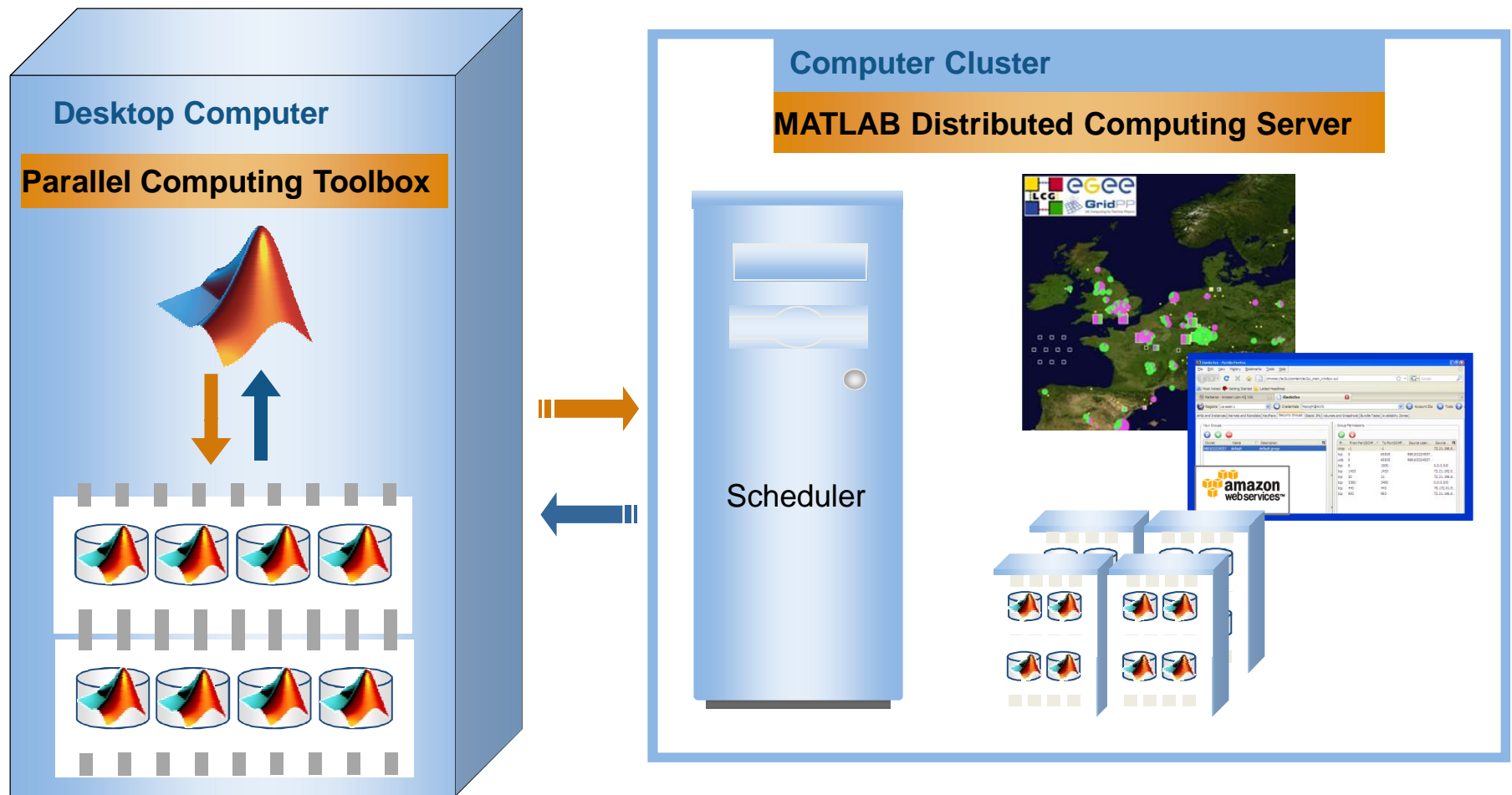
Data-Parallel

Large data problems

11	26	41
12	27	42
13	28	43
14	29	44
15	30	45
16	31	46
17	32	47
17	33	48
19	34	49
20	35	50
21	36	51
22	37	52



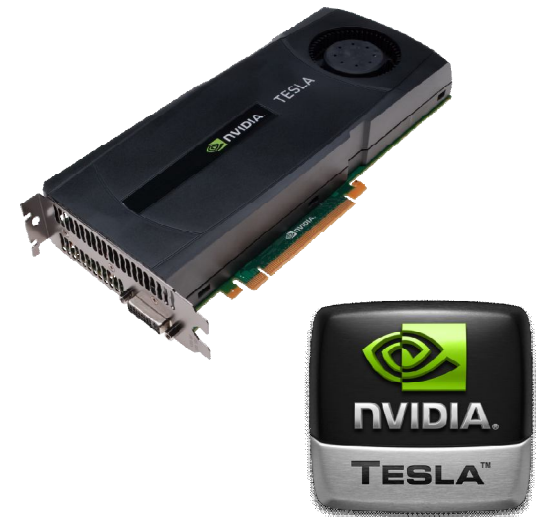
Parallel and Distributed Computing Products



What is a Graphics Processing Unit (GPU)

- Originally for graphics acceleration, now also used for scientific calculations
- Massively parallel array of integer and floating point processors
 - Typically hundreds of processors per card
 - GPU cores complement CPU cores
- Dedicated high-speed memory

**New: GPU
functionality in
R2010b**



* Parallel Computing Toolbox requires NVIDIA GPUs with Compute Capability 1.3 or greater, including NVIDIA Tesla 10-series and 20-series products. See http://www.nvidia.com/object/cuda_gpus.html for a complete listing

Summary of Options for Targeting GPUs

Ease of Use

- 1) Use GPU array interface with MATLAB built-in functions
- 2) Execute custom functions on elements of the GPU array
- 3) Invoke your CUDA kernels directly from MATLAB

Greater Control

Additional Resources

On-Demand Recorded Webinars:

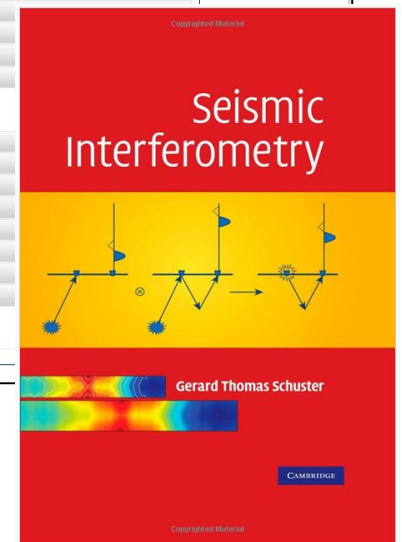
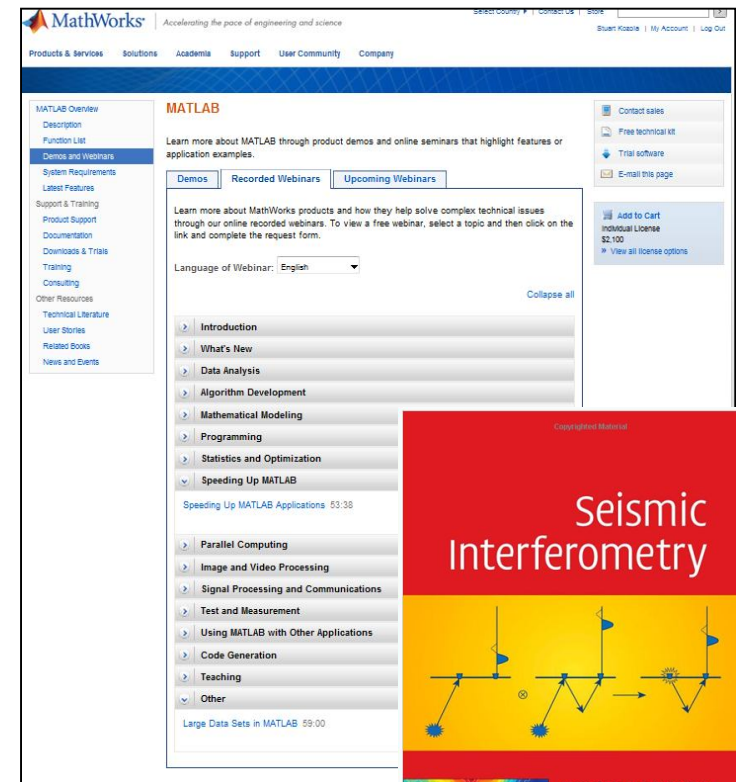
- Large Data in MATLAB
- Speeding up MATLAB Applications

MathWorks Services

- Training
- Consulting

Partner Programs/User Community

- Book program
- 3rd party solutions
- MATLAB Central



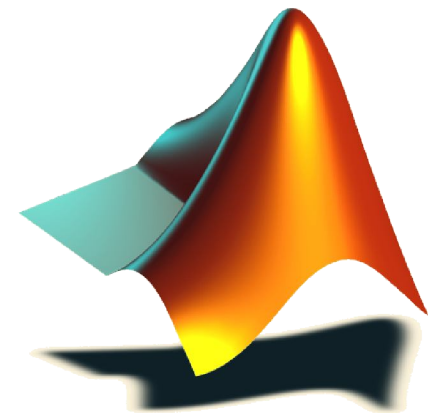
Contact Information

North America

- Phone: 508-647-7000
- E-mail: support@mathworks.com

Outside North America

- Contact your local MathWorks office or reseller:
www.mathworks.com/contact



Questions?