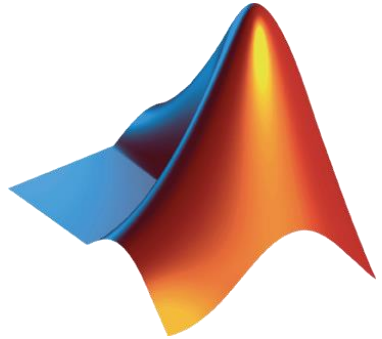
Parallel and Distributed Computing with MATLAB



© 2018 The MathWorks, Inc.

Practical Application of Parallel Computing

- Why parallel computing?
  - Need faster insight on more complex problems with larger datasets
  - 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

Steps for Improving Performance

- First get code working
- Speed up code with core MATLAB
- Include compiled languages and additional hardware

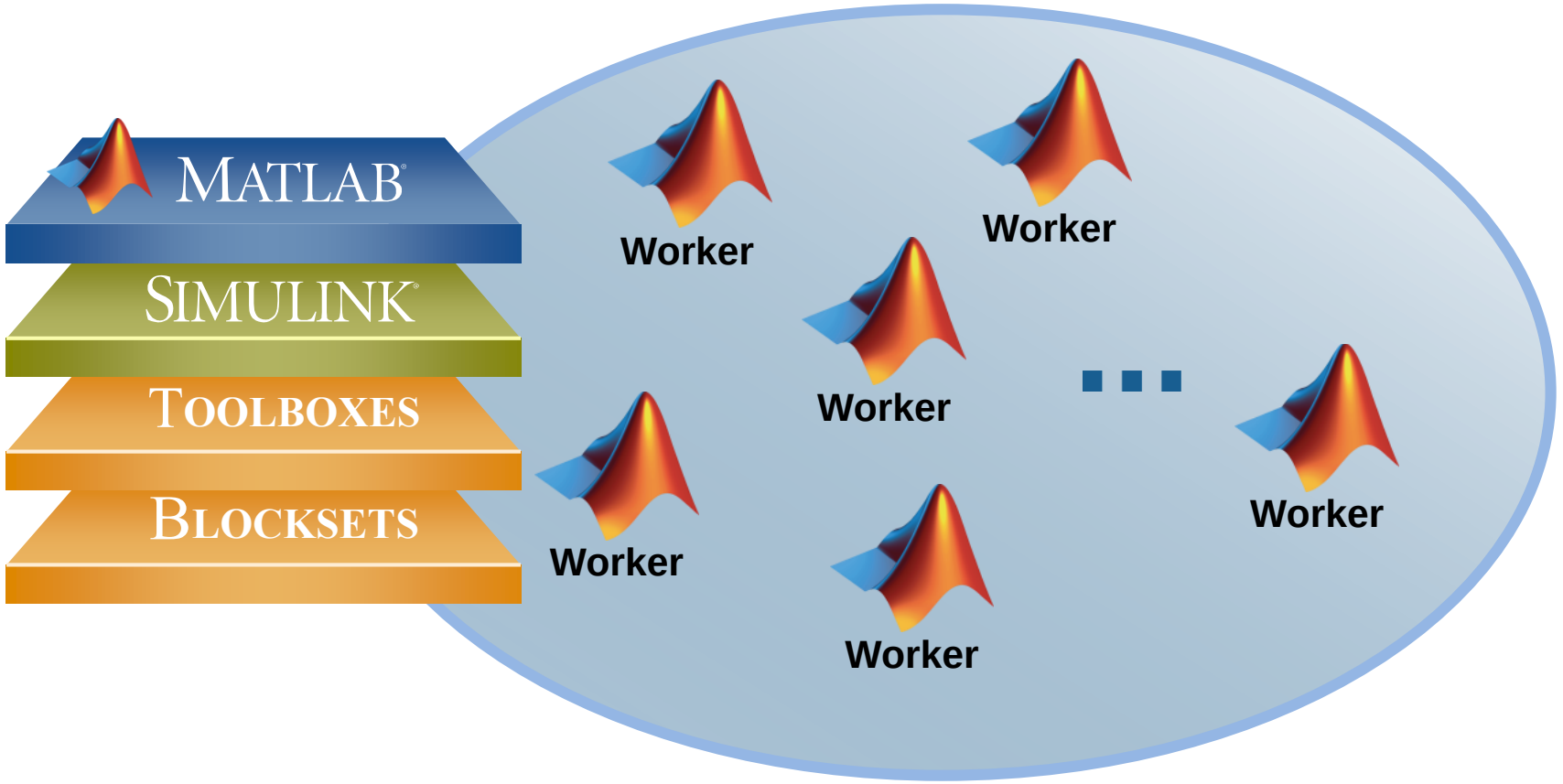
Webinar: Optimizing and Accelerating Your MATLAB Code

Programming Parallel Applications

- Built-in multithreading
  - Automatically enabled in MATLAB since R2008a
  - Multiple threads in a single MATLAB computation engine

- Parallel computing using explicit techniques
  - Multiple computation engines controlled by a single session
  - High-level constructs to let you parallelize MATLAB applications
  - Perform MATLAB computations on GPUs

Parallel Computing

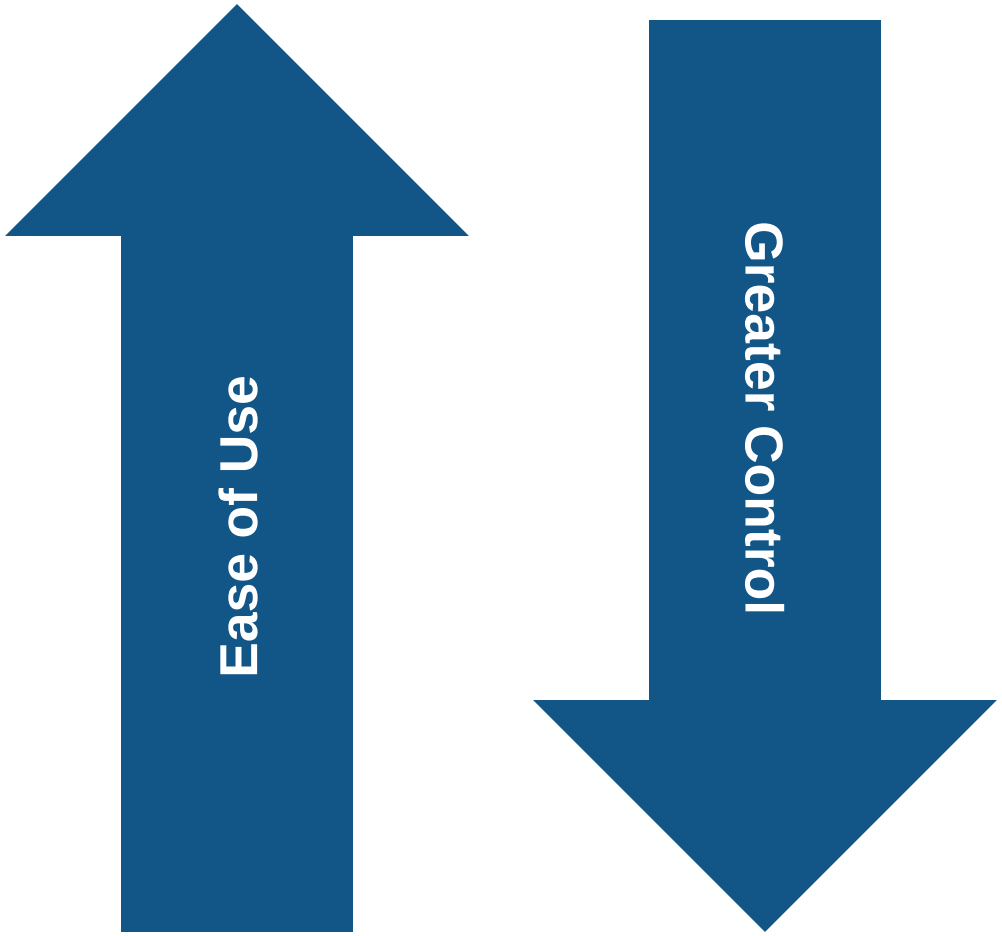


Agenda

- Utilizing multiple cores on a desktop computer
- Scaling up to cluster and cloud resources
- Tackling data-intensive problems on desktops and clusters
- Accelerating applications with NVIDIA GPUs
- Summary and resources

Programming Parallel Applications

- Built in support
  - `..., 'UseParallel', true)`



Parallel-enabled Toolboxes

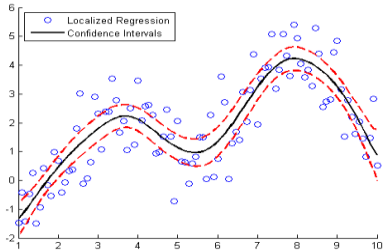
Image Processing

Batch Image Processor, Block Processing, GPU-enabled functions



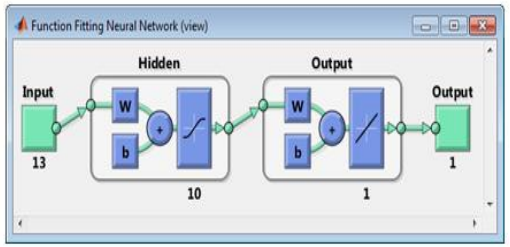
Statistics and Machine Learning

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



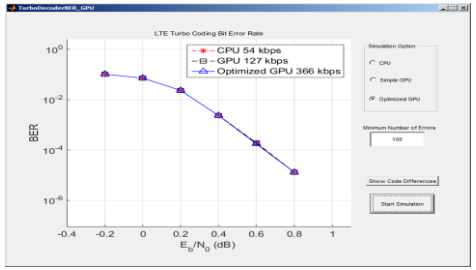
Neural Networks

Deep Learning, Neural Network training and simulation



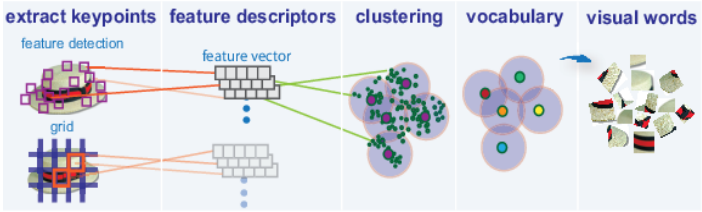
Signal Processing and Communications

GPU-enabled FFT filtering, cross correlation, BER



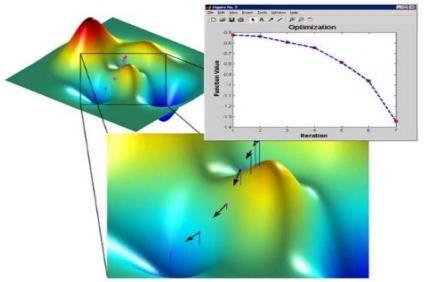
Computer Vision

Parallel-enabled functions in bag-of-words workflow



Optimization

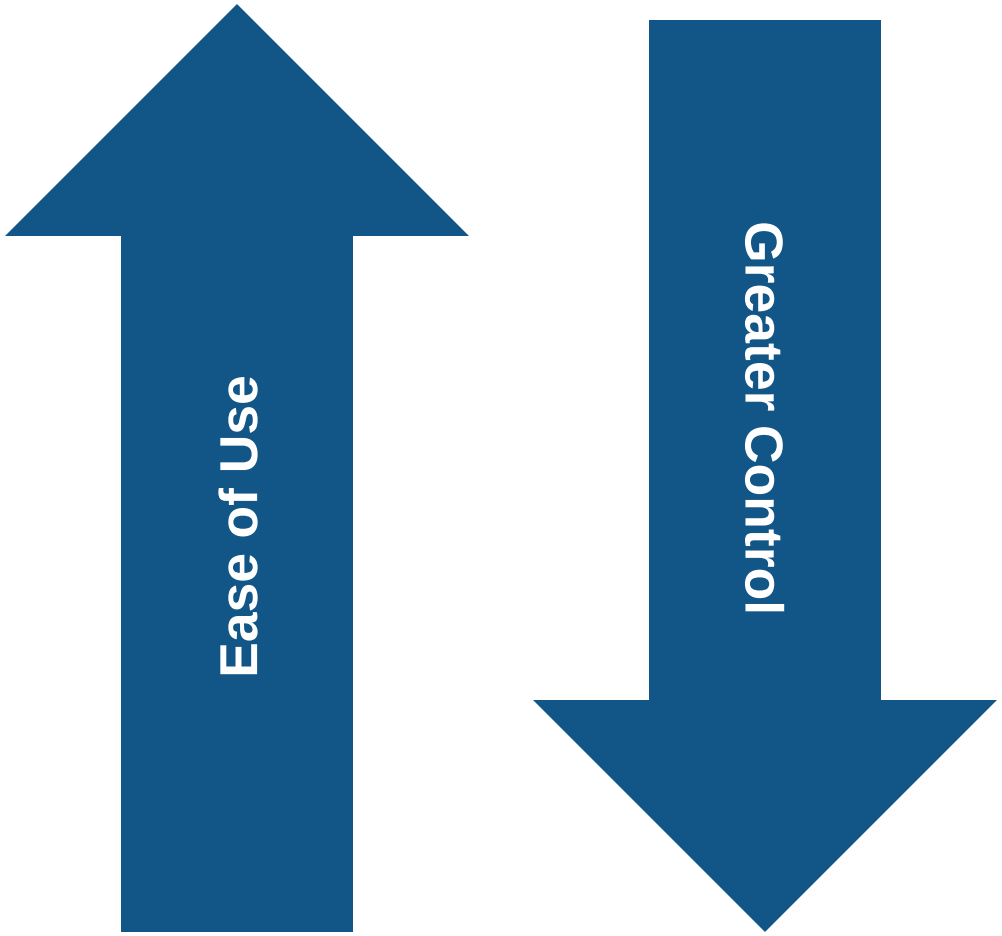
Parallel estimation of gradients



[Other Parallel-enabled Toolboxes]

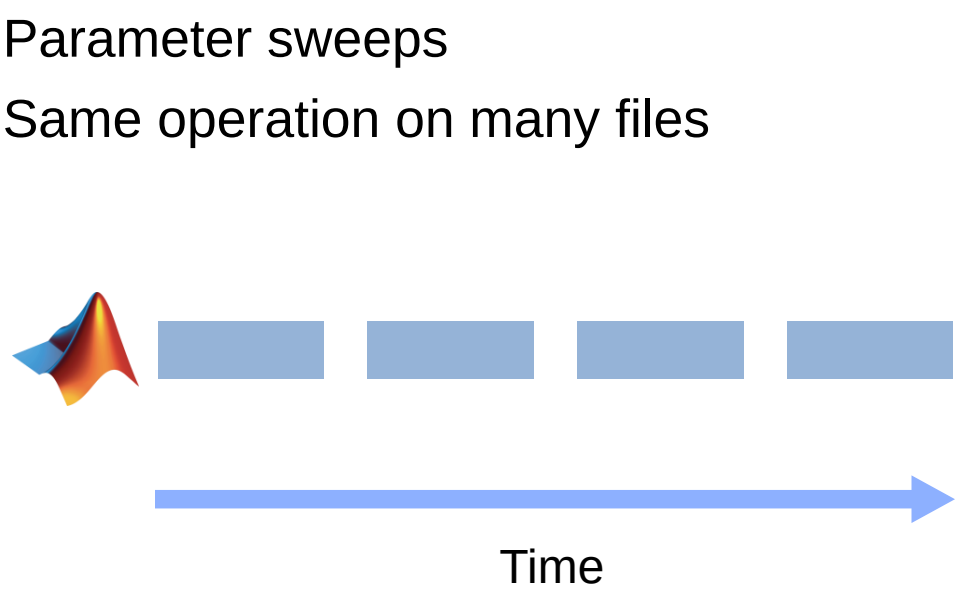
Programming Parallel Applications

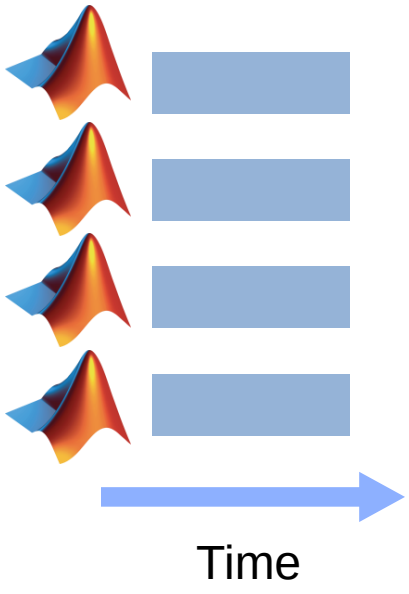
- Built in support
  - `..., 'UseParallel', true)`
- Simple programming constructs
  - `parfor`, `batch`



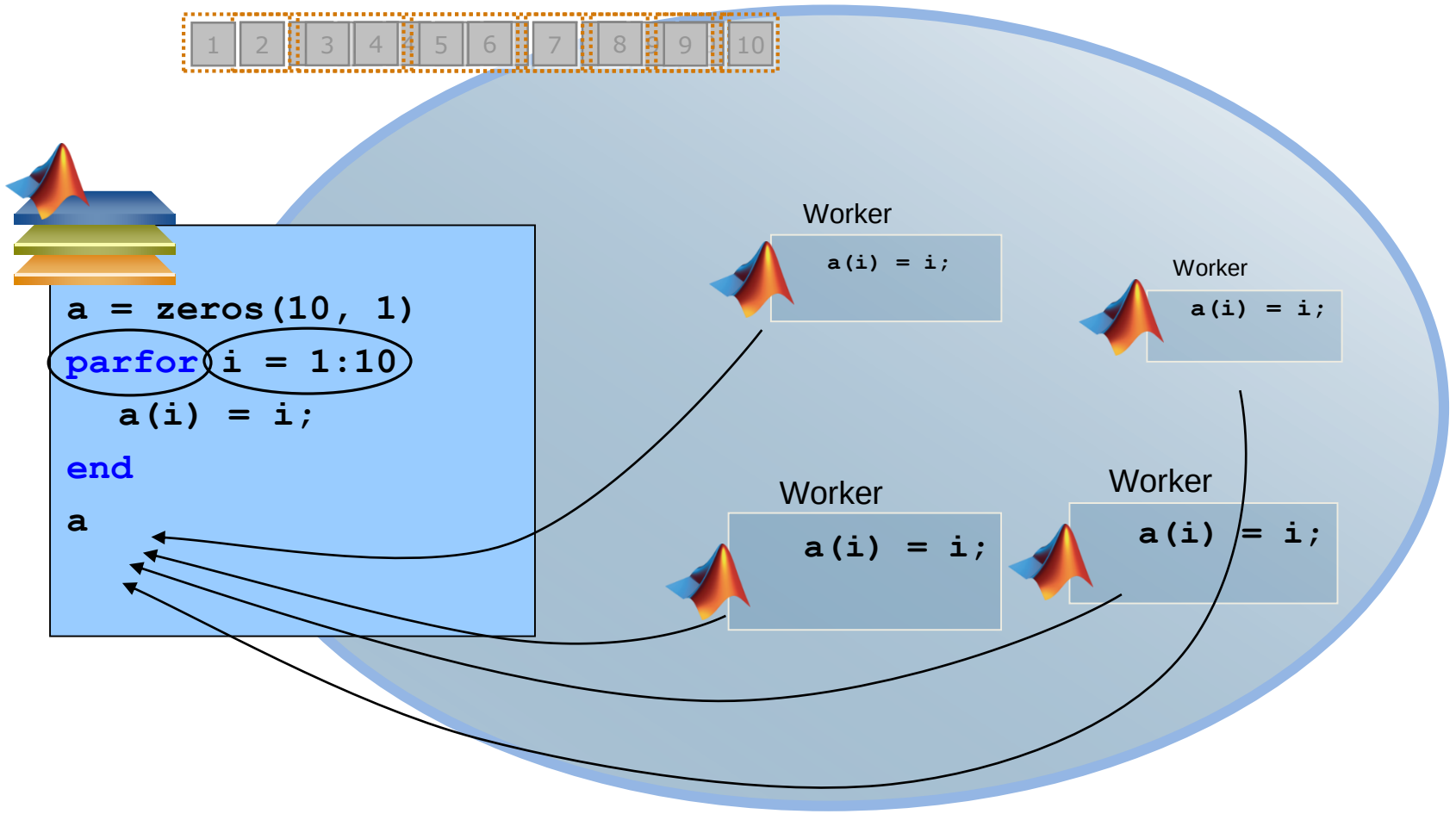
Embarrassingly Parallel: Independent Tasks or Iterations

- No dependencies or communication between tasks
- Examples:
  - Monte Carlo simulations
  - Parameter sweeps
  - Same operation on many files





Mechanics of `parfor` Loops



Tips for Leveraging PARFOR

- Consider creating smaller arrays on each worker versus one large array prior to the parfor loop

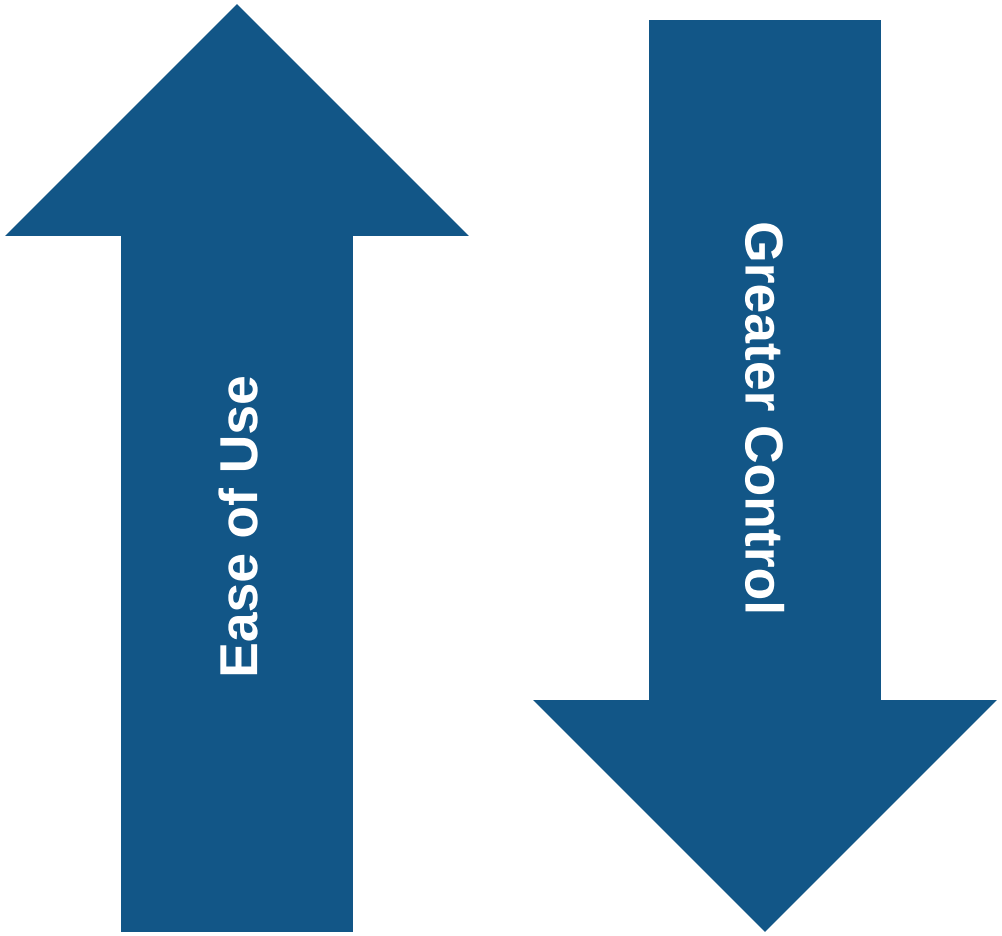
- Take advantage of `parallel.pool.Constant` to establish variables on pool workers prior to the loop

- Encapsulate blocks as functions when needed

Understanding parfor

Programming Parallel Applications

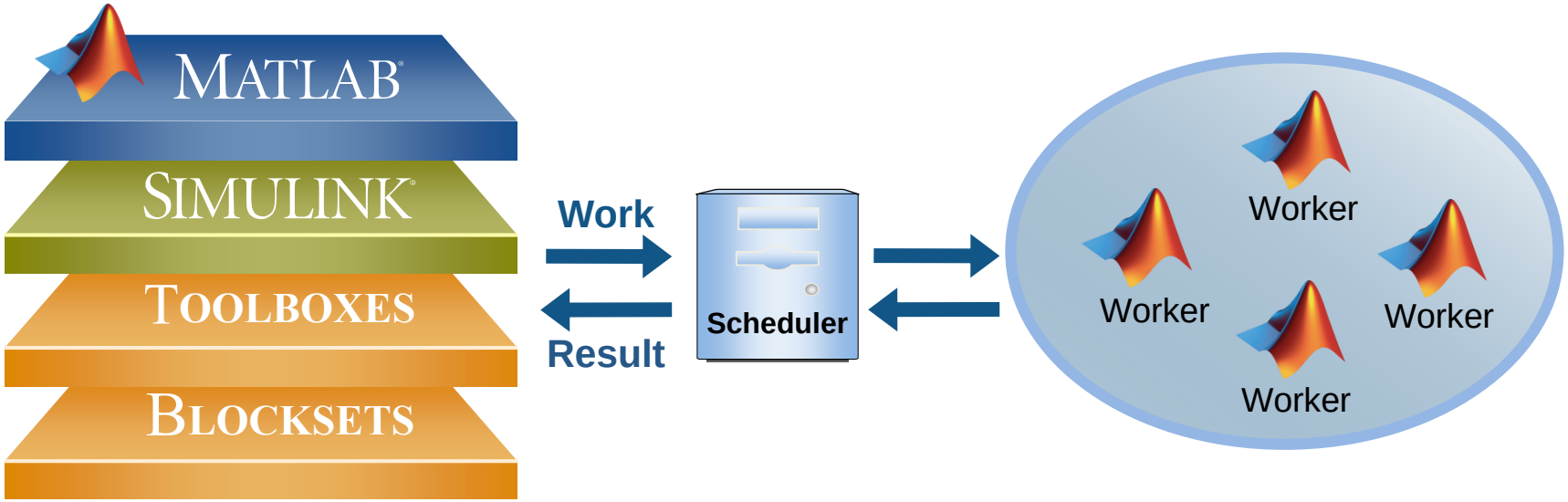
- Built in support
  - `..., 'UseParallel', true)`
- Simple programming constructs
  - `parfor, batch`
- Full control of parallelization
  - `spmd, parfeval`



Agenda

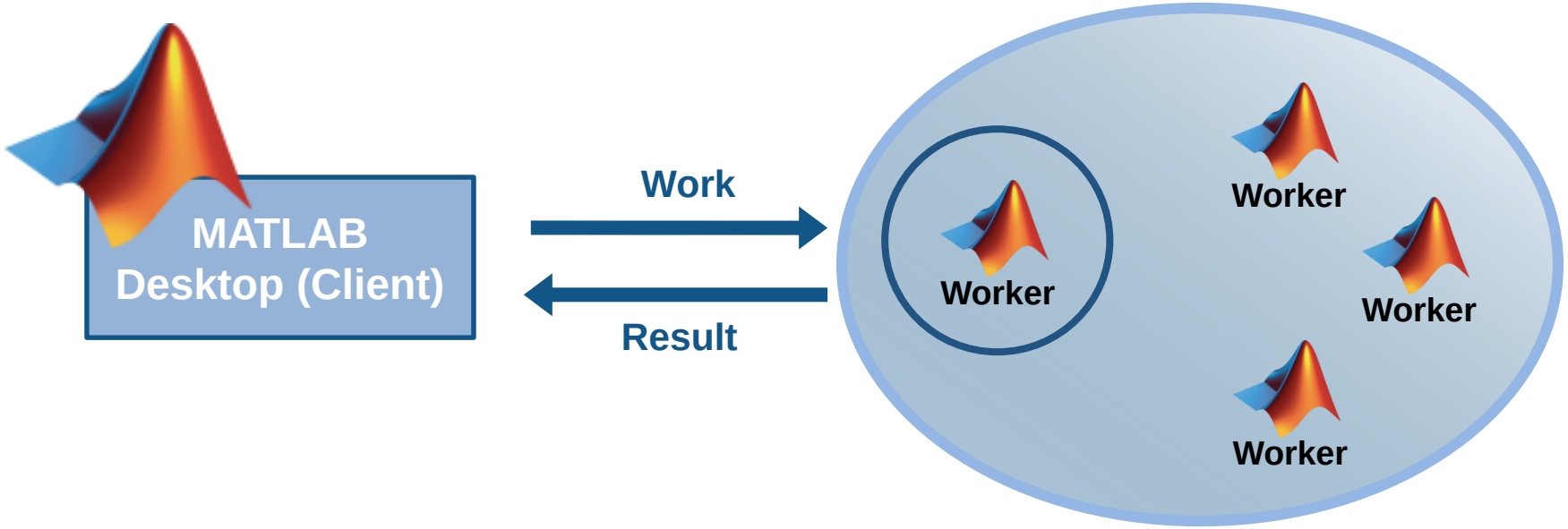
- Utilizing multiple cores on a desktop computer
- Scaling up to cluster and cloud resources
- Tackling data-intensive problems on desktops and clusters
- Accelerating applications with NVIDIA GPUs
- Summary and resources

Offloading Computations



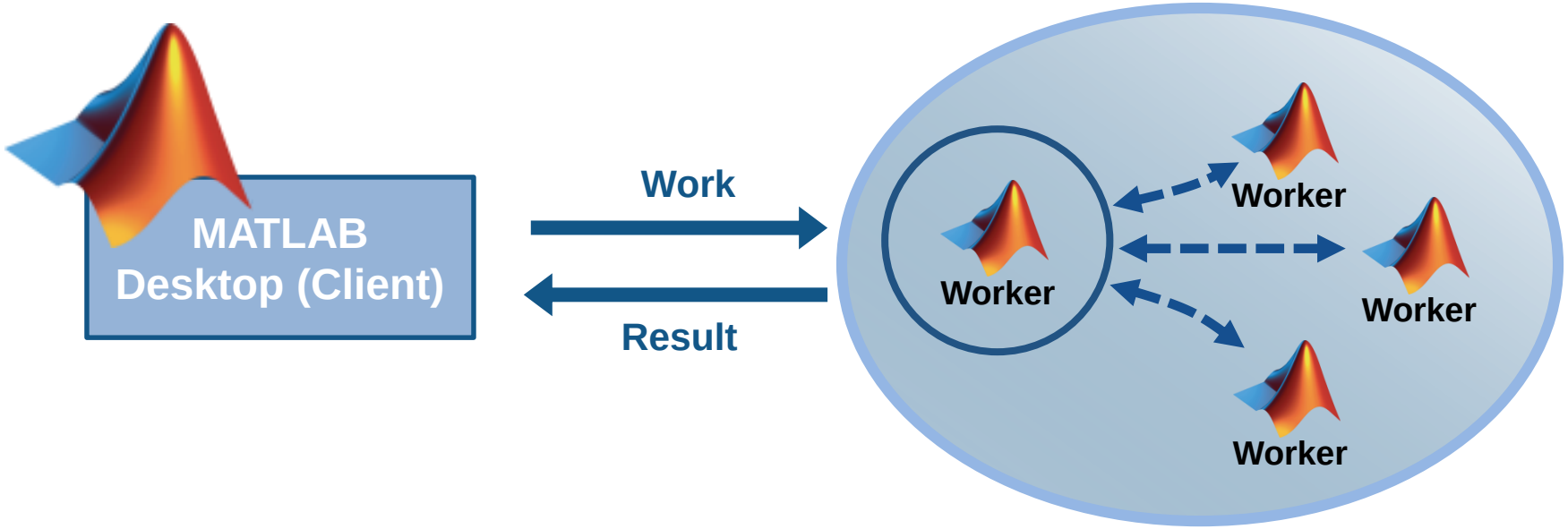
Offloading Serial Computations

- `job = batch(...);`



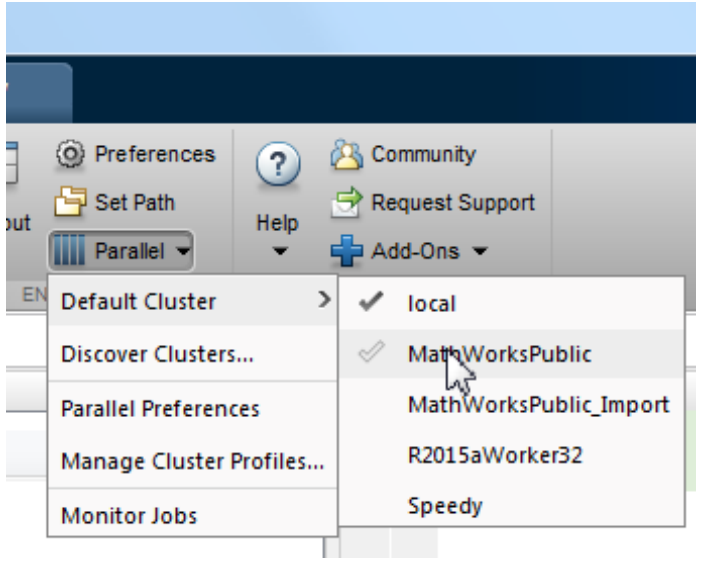
Offloading and Scaling Computations

- `job = batch(... , 'Pool', n);`



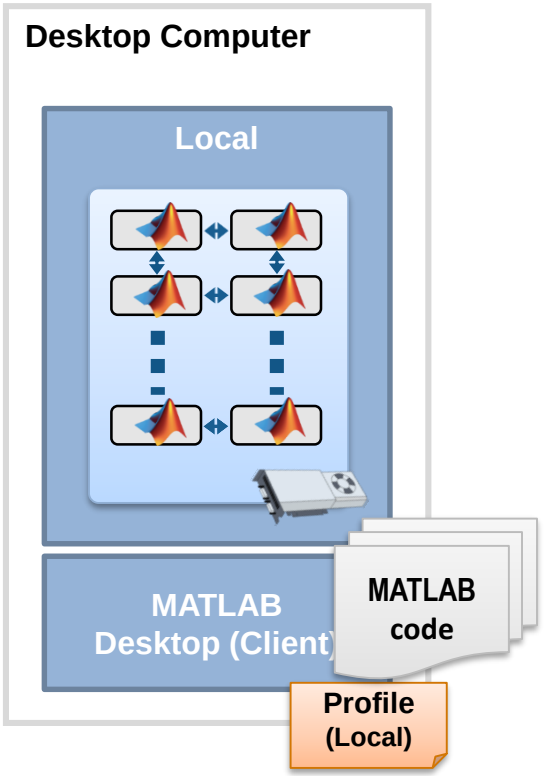
Migrate to Cluster / Cloud

- Use MATLAB Distributed Computing Server
- Change hardware without changing algorithm



Use MATLAB Distributed Computing Server

1. Prototype code



以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：https://d.book118.com/195312021020011134