The MathWorks MATLAB® & SIMULINK®

Hands-On Session for Parallel Computing with MATLAB® and gLite

Elwin Chan

Arjav Chakravarti

Solving Big Technical Problems

Difficulties	You could…		Solutions
Long running Computationally intensive	Wait	⇒	Run similar *tasks* on independent processors in *parallel*
Large data set	Reduce size of problem	⇒	Load *data* onto multiple machines that work together in *parallel*

Parallel Computing

Difficulties

Jobs run in scheduled mode

Hard to debug

Cannot access intermediate answers

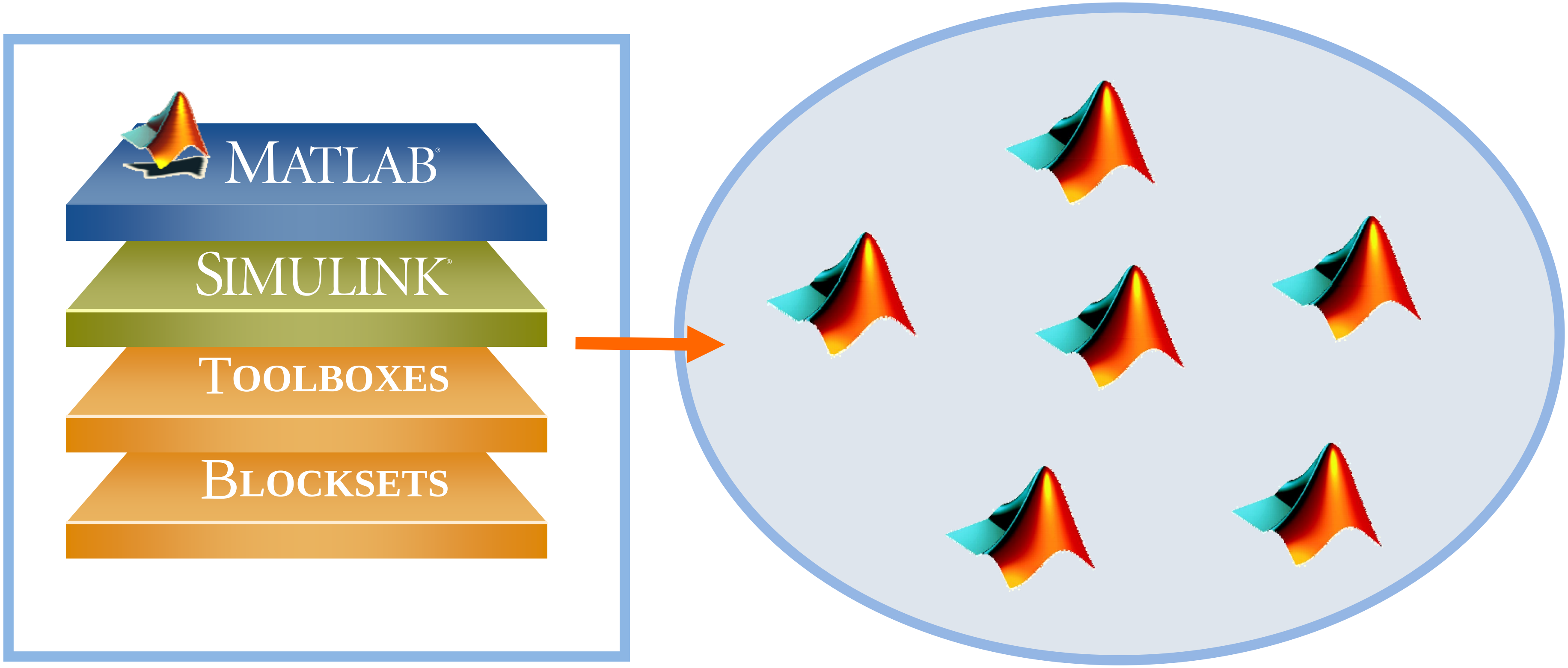
Hard to diagnose bottlenecks in algorithm

⇨

Solution

Work *interactively* in parallel

Parallel Computing with MATLAB



Pool of MATLAB Workers

Parallel Computing with MATLAB™

No code changes

- Implicit Multithreaded MATLAB™
- Toolbox Support:

  Optimization Toolbox™

  Genetic Algorithm and Direct Search Toolbox™

  SystemTest™

Task Parallel

Data Parallel

Trivial changes

- `parfor`
- `distributed`
- `job` and `tasks`
- MATLAB and MPI

Extensive changes

Agenda

Speed up algorithms without code changes

- Develop parallel code interactively
  - Task parallel applications for faster processing
  - Data parallel applications for handling large data sets
- Schedule your programs to run

Parallel Computing with MATLAB™

No code changes

- Implicit Multithreaded MATLAB™
- Toolbox Support:
  - Optimization Toolbox™
  - Genetic Algorithm and Direct Search Toolbox™
  - SystemTest™

Task Parallel ***Data Parallel***

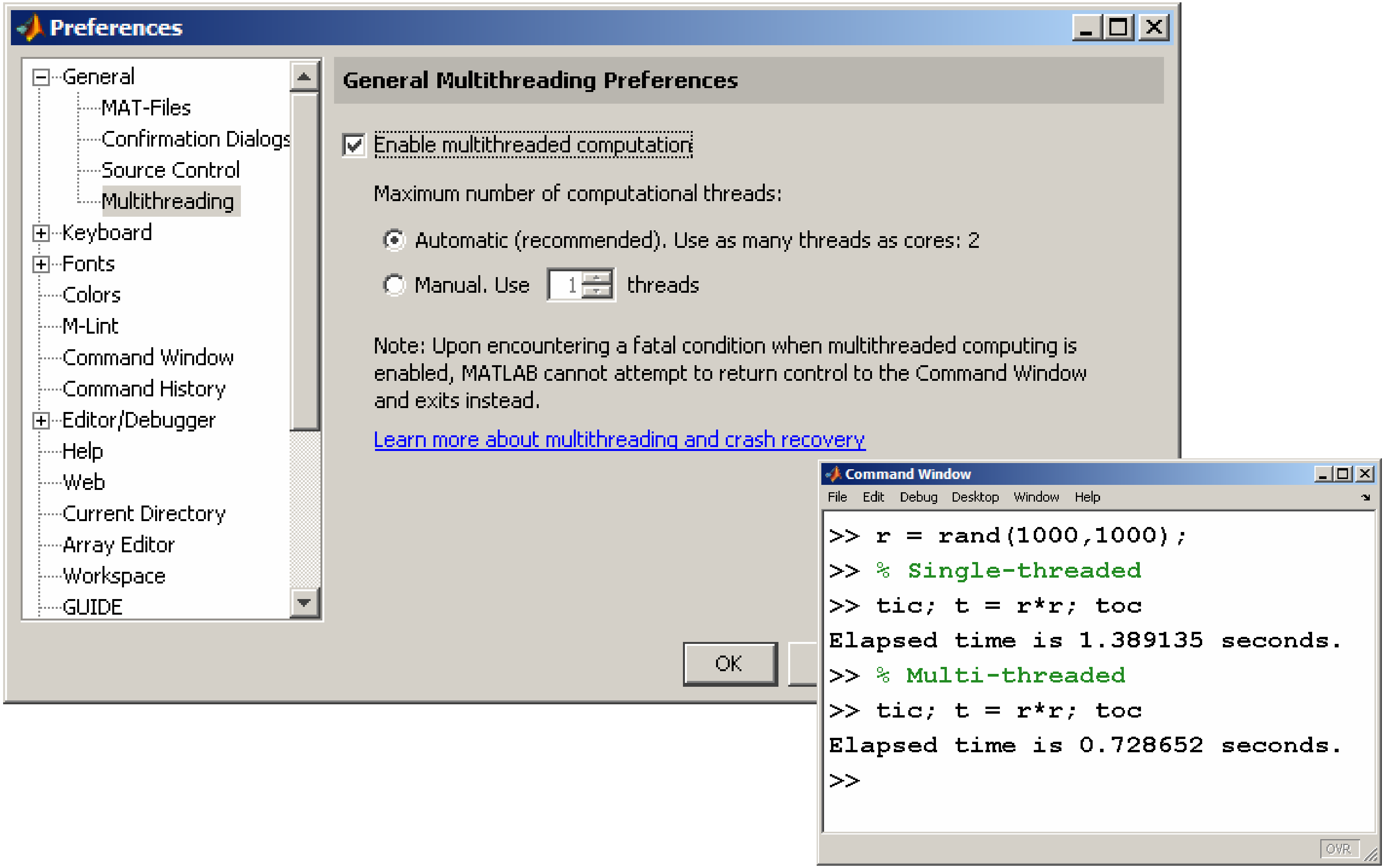
Trivial changes

- `parfor`
- `distributed`
- `job` and `tasks`

Extensive changes

- MATLAB and MPI

Exercise 1: Speed up mathematical operations



General Multithreading Preferences

☑ Enable multithreaded computation

Maximum number of computational threads:

● Automatic (recommended). Use as many threads as cores: 2

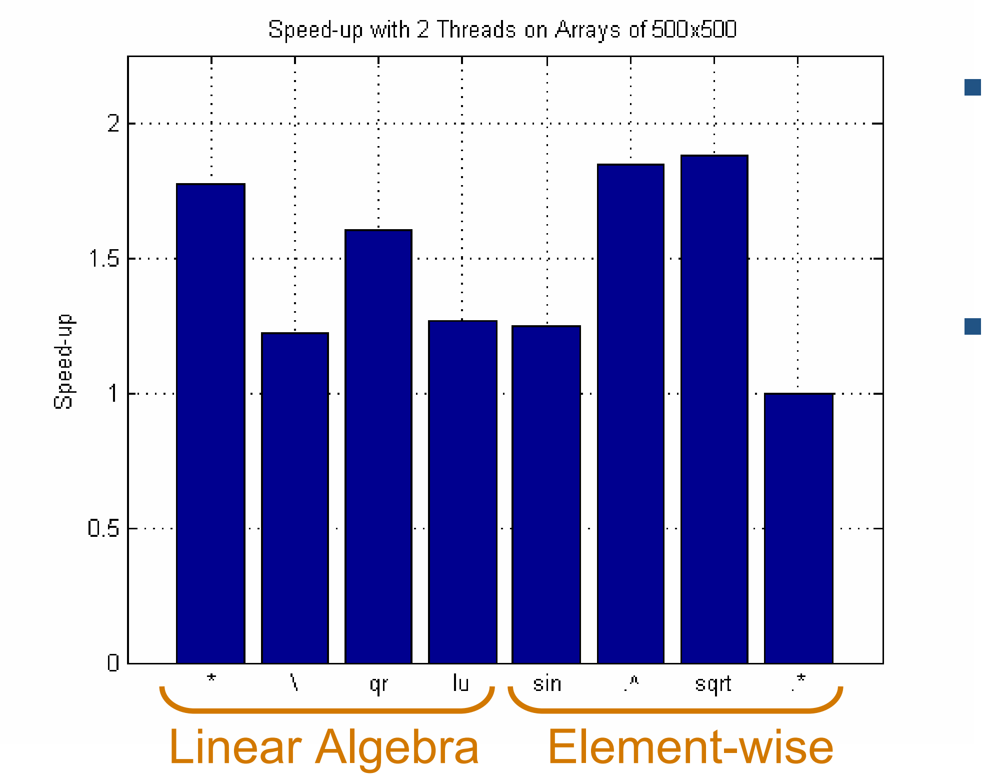
○ Manual. Use 1 threads

Note: Upon encountering a fatal condition when multithreaded computing is enabled, MATLAB cannot attempt to return control to the Command Window and exits instead.

Learn more about multithreading and crash recovery

```
>> r = rand(1000,1000);
>> % Single-threaded
>> tic; t = r*r; toc
Elapsed time is 1.389135 seconds.
>> % Multi-threaded
>> tic; t = r*r; toc
Elapsed time is 0.728652 seconds.
>>
```

Example 1: Speed up for Implicit Multithreaded Computations



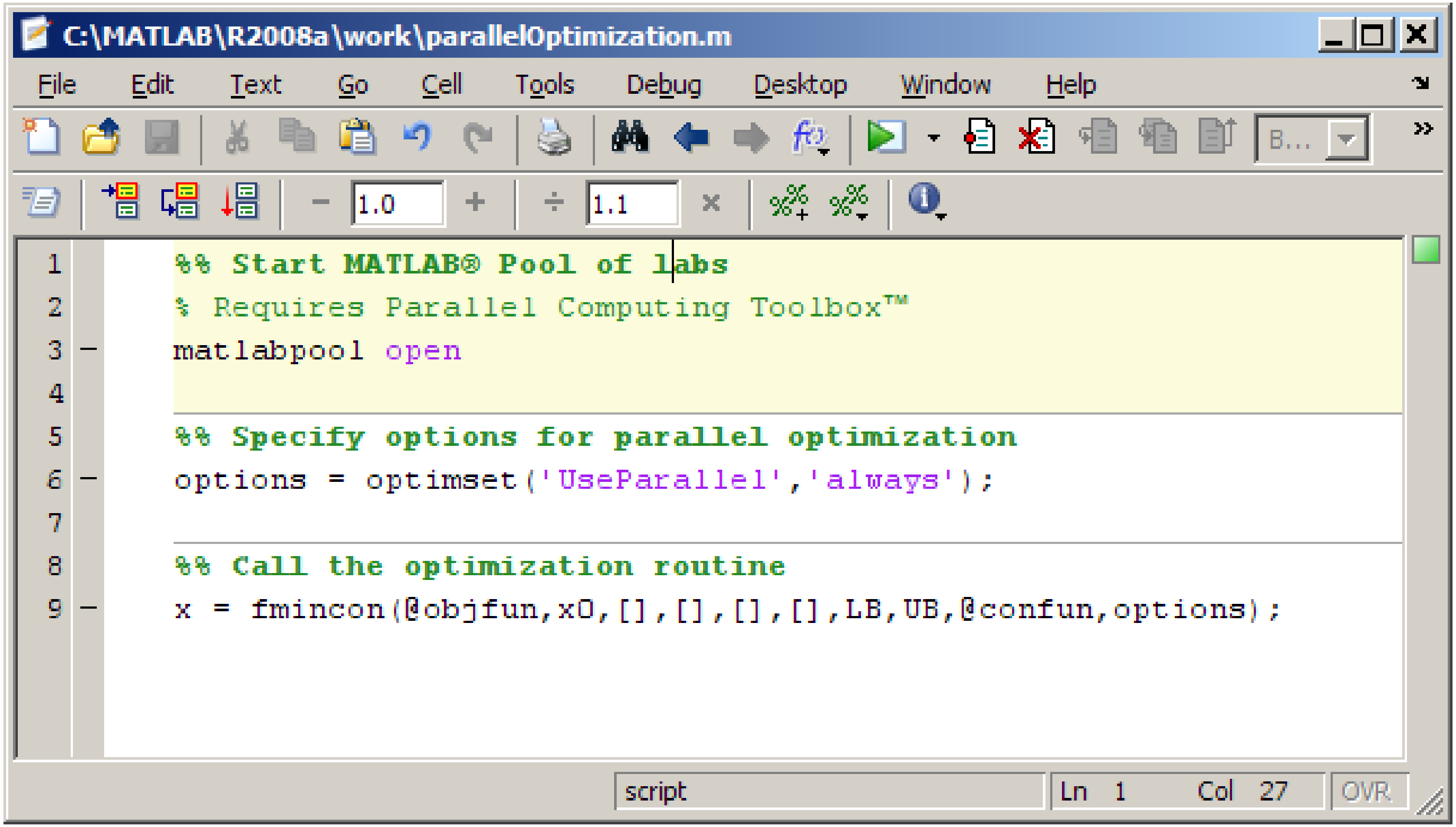
- No change required for user code
- Enables multithreading for key mathematical routines
  - Linear Algebra operations
  - Element-wise operations

Implicit Multithreaded Computation

- Linear Algebra Operations
  - Uses multithreaded Basic Linear Algebra Subroutines (BLAS)
  - BLAS are vendor specific
  - Optimized for specific processor
- Element-Wise Operations
  - Just-In-Time acceleration (JIT) generates on-the-fly multithreaded code

Support in Optimization Toolbox™

New in R2008a



```
%% Start MATLAB® Pool of labs
% Requires Parallel Computing Toolbox™
matlabpool open

%% Specify options for parallel optimization
options = optimset('UseParallel','always');

%% Call the optimization routine
x = fmincon(@objfun,x0,[],[],[],[],LB,UB,@confun,options);
```

Agenda

- Speed up algorithms without code changes
- Develop parallel code interactively
  - Task parallel applications for faster processing
  - Data parallel applications for handling large data sets
- Schedule your programs to run

Parallel Computing with MATLAB™

No code changes

- Implicit Multithreaded MATLAB™
- Toolbox Support:

  Optimization Toolbox™

  Genetic Algorithm and Direct Search Toolbox™

  SystemTest™

Task Parallel

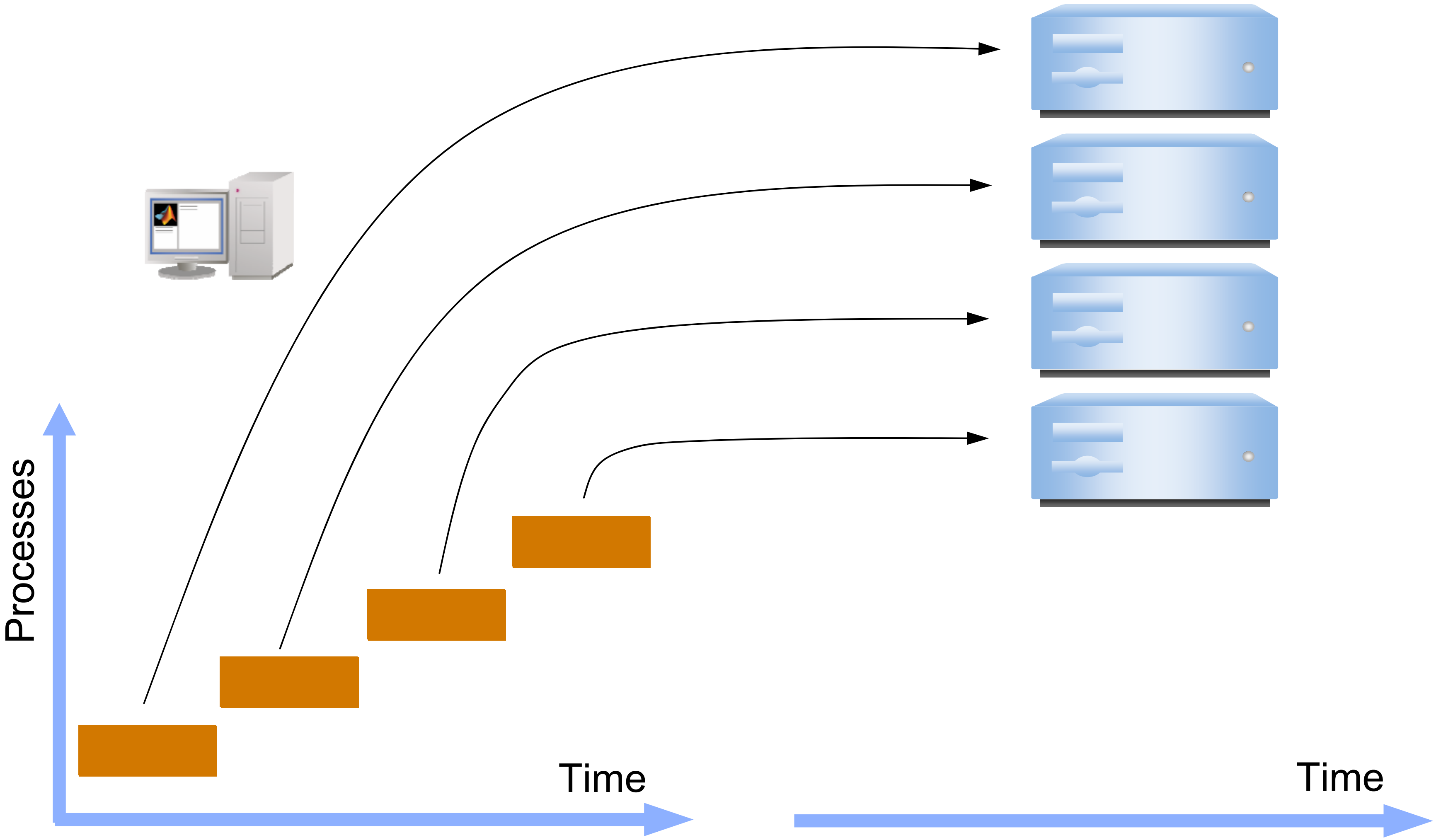
Data Parallel

Trivial changes

- `parfor`
- `distributed`
- `job` and `tasks`
- MATLAB and MPI

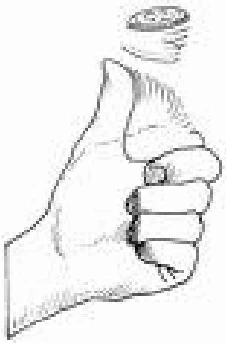
Extensive changes

Distributing Tasks (Task Parallel)



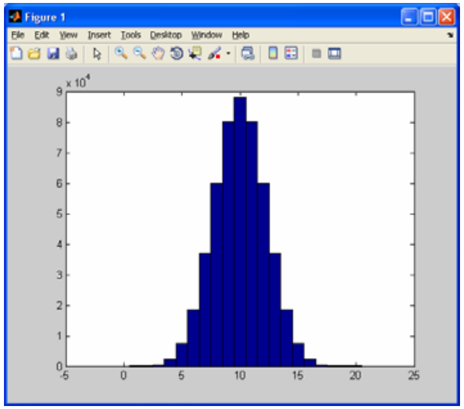
Example 3: Monte Carlo Simulation of Coin Tossing

10 Simulations of Flipping 20 Coins at a Time



11	7	12	15	7	9	9	7	8	12

Number of Heads out of 20



Example 4,5: Monte Carlo Simulation of Coin Tossing

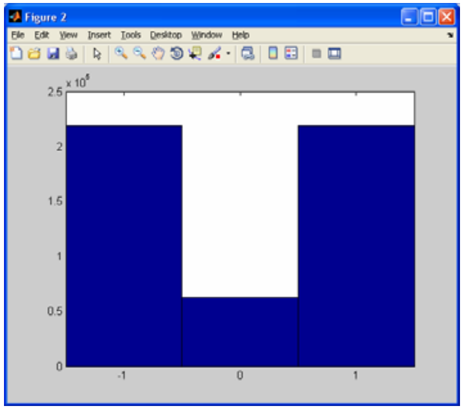
10 Simulations of Flipping 20 Coins at a Time

11	7	12	15	9	9	13	7	8	12

Number of Heads out of 20

Change in Number of Heads

--	↓	↑	↑	↓	→	↑	↓	↑	↑



Parallel for loops

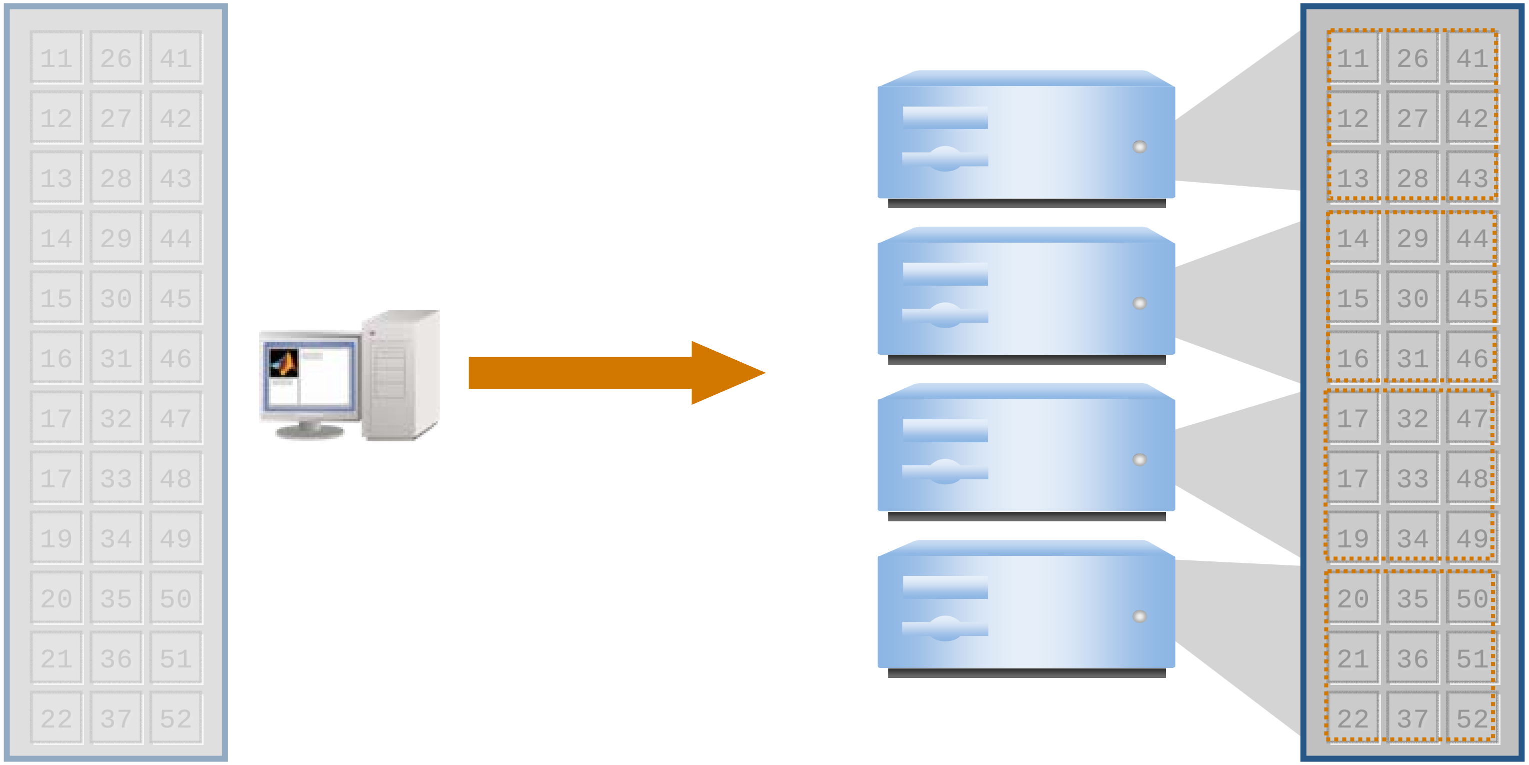
```
parfor i = 1 : n
   % do something with i
end
```

- Mix task parallel and serial code in the same function
- Run loops on a pool of MATLAB resources
- Iterations must be order-independent
- M-Lint analysis helps in converting existing for loops into to parfor loops

Agenda

- Speed up algorithms without code changes
- Develop parallel code interactively
  - Task parallel applications for faster processing
  - Data parallel applications for handling large data sets
- Schedule your programs to run

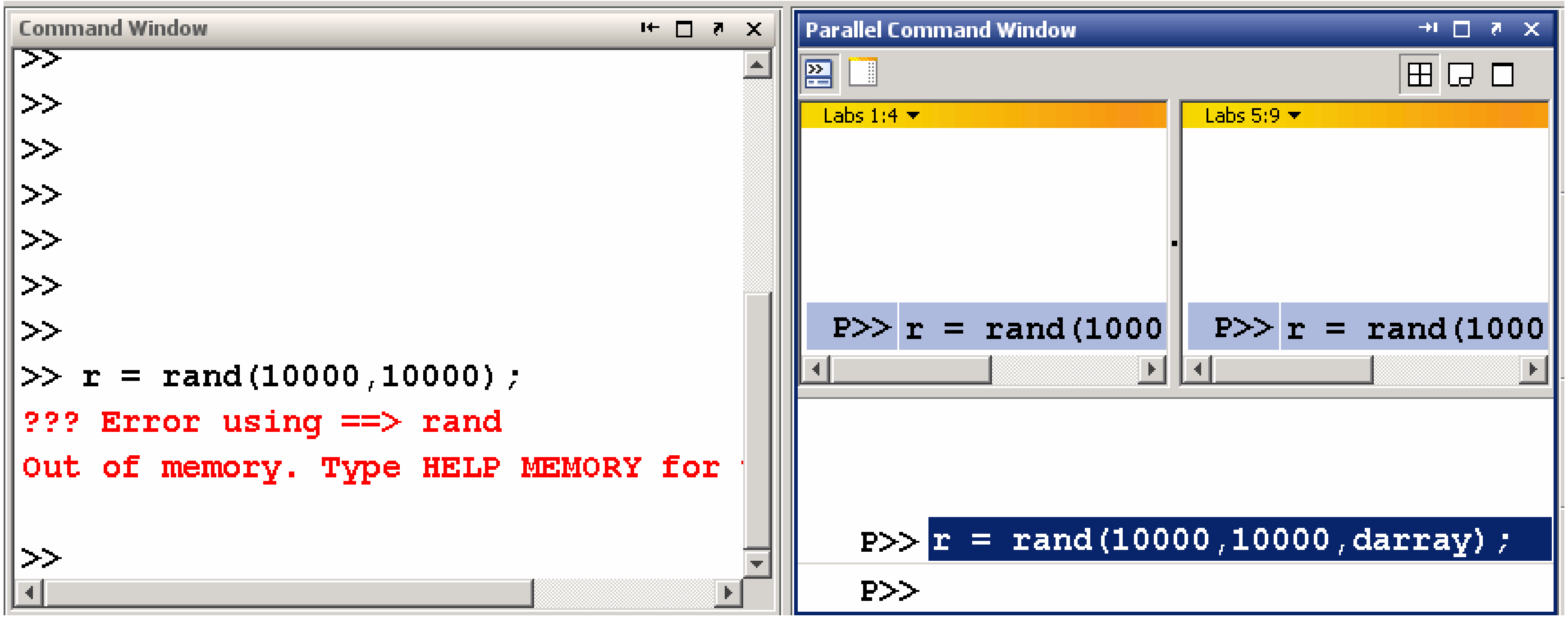
Large Data Sets (Data Parallel)



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

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

Example 6: Parallel Mode (pmode) and Distributing Arrays



```
>>
>>
>>
>>
>>
>>
>>
>> r = rand(10000,10000);
??? Error using ==> rand
Out of memory. Type HELP MEMORY for

>>
```

```
P>> r = rand(1000
```

```
P>> r = rand(1000
```

```
P>> r = rand(10000,10000,darray);
P>>
```

Distributed Arrays, Parallel Algorithms

- Distributed arrays
  - Store segments of data across participating workers
  - Create from any built-in class in MATLAB
    - Examples: doubles, sparse, logicals, cell arrays, and arrays of structs
- Parallel algorithms for distributed arrays
  - Matrix manipulation operations
    - Examples: indexing, data type conversion, and transpose
  - Parallel linear algebra functions such as **`svd`** and **`lu`**
  - Data distribution
    - Automatic, specify your own, or change at any time

MPI-Based Functions in Parallel Computing Toolbox™

Use when a high degree of control over parallel algorithm is required

- High-level abstractions of MPI functions
  - **`labSendReceive`**, **`labBroadcast`**, and others
  - Send, receive, and broadcast any data type in MATLAB

- Automatic bookkeeping
  - Setup: communication, ranks, etc.
  - Error detection: deadlocks and miscommunications

- Pluggable
  - Use any MPI implementation that is *binary*-compatible with MPICH-2

Parallel Computing with MATLAB™

No code changes

- Implicit Multithreaded MATLAB™
- Toolbox Support:

  Optimization Toolbox™

  Genetic Algorithm and Direct Search Toolbox™

  SystemTest™

Task Parallel

Data Parallel

Trivial changes

- `parfor`
- `job` and `tasks`
- `distributed`

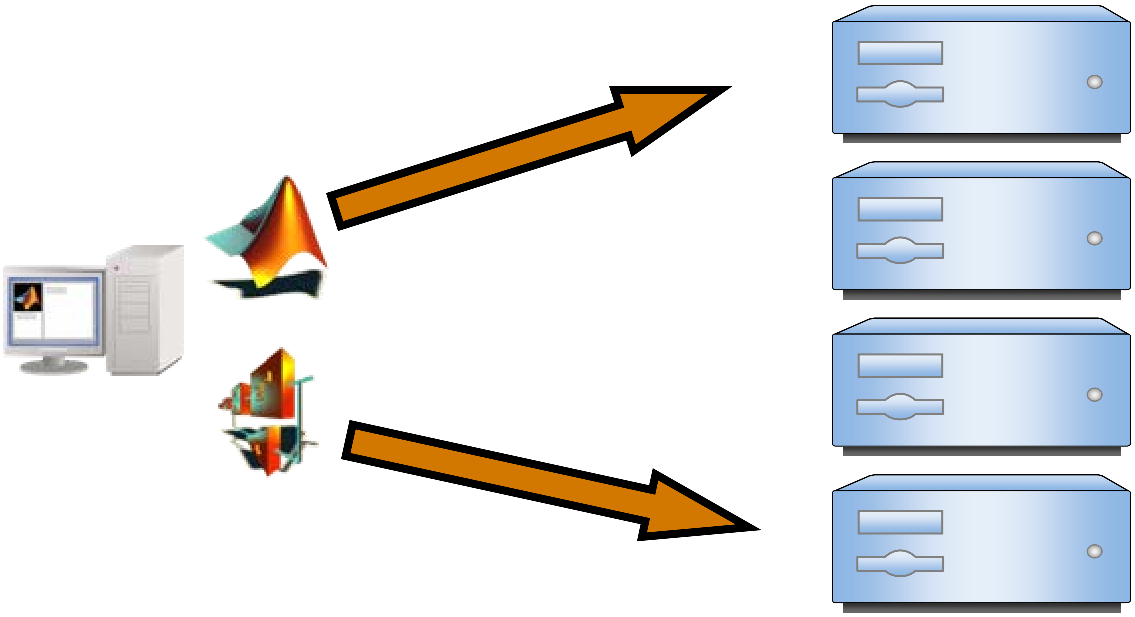
Extensive changes

- MATLAB and MPI

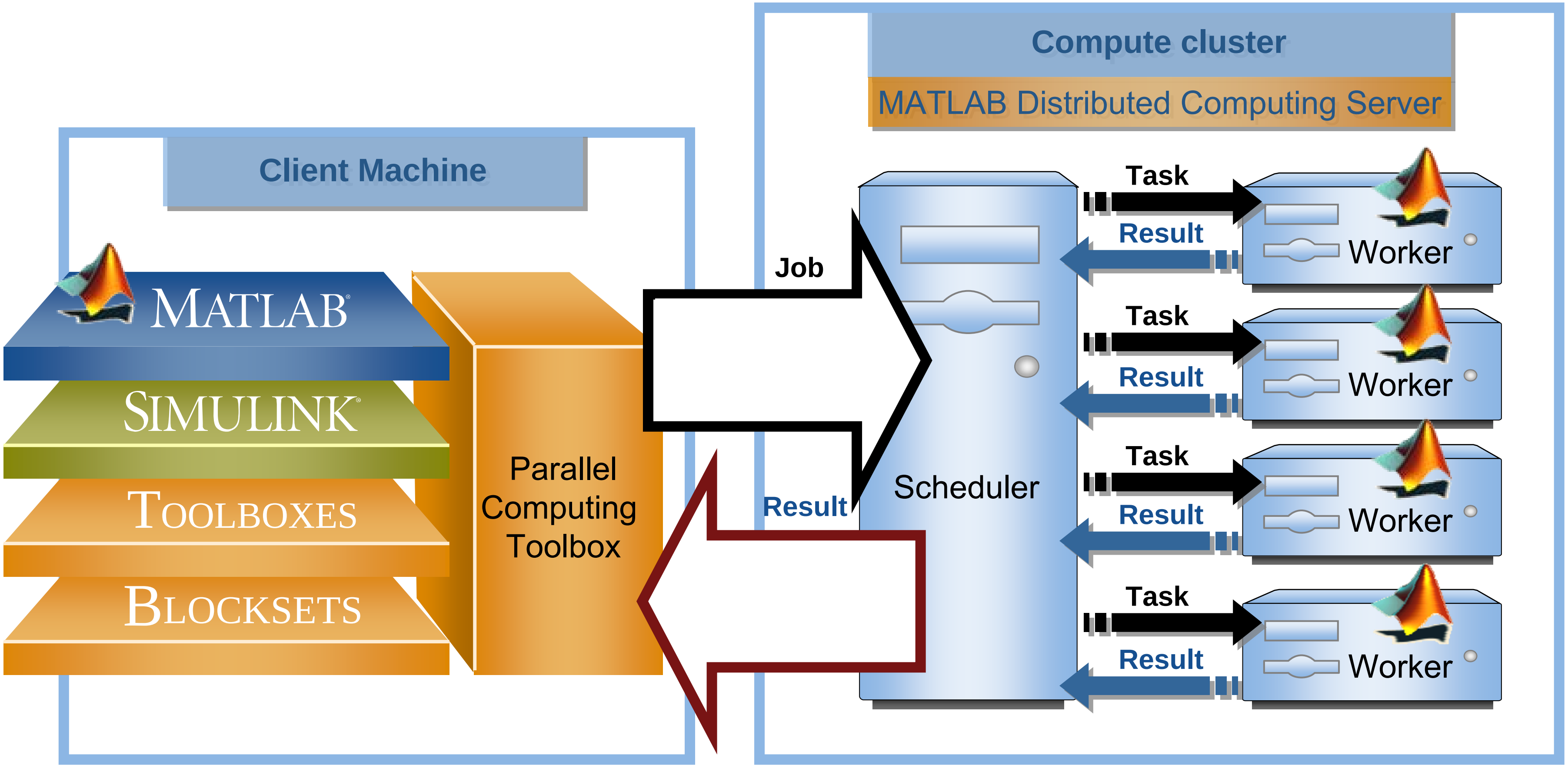
Agenda

- Speed up algorithms without code changes
- Develop parallel code interactively
  - Task parallel applications for faster processing
  - Data parallel applications for handling large data sets
- Schedule your programs to run

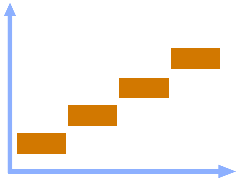
Off-loading from your machine

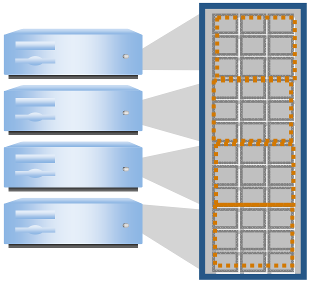


Distributed applications



Options for Scheduling Jobs





Task Parallel	Data Parallel
>> createMatlabPoolJob or >> batch New in R2008a	>> createParallelJob
>> createJob(…) >> createTask(…)	

Dependencies

- `job – FileDependencies`
  - Files are copied from client to each worker machine
  - ZIP compressed
  - Uncompressed and added to the MATLAB path
  - Convenient for .m files, but can be slow for large data files
- `job – PathDependencies`
  - Shared directories are added to the MATLAB path
  - Mixing of Windows and UNIX paths allowed
  - Reduces the amount of data transfer from client to cluster

Summary

- Speed up algorithms without code changes
- Develop parallel code interactively
  - Task parallel applications for faster processing
  - Data parallel applications for handling large data sets
- Schedule your programs to run