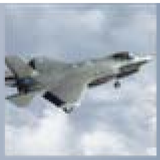
The MathWorks

MATLAB® & SIMULINK®

Distributed Computing in the Engineering Workflow

Loren Dean
Director of Engineering, MATLAB Products
The MathWorks

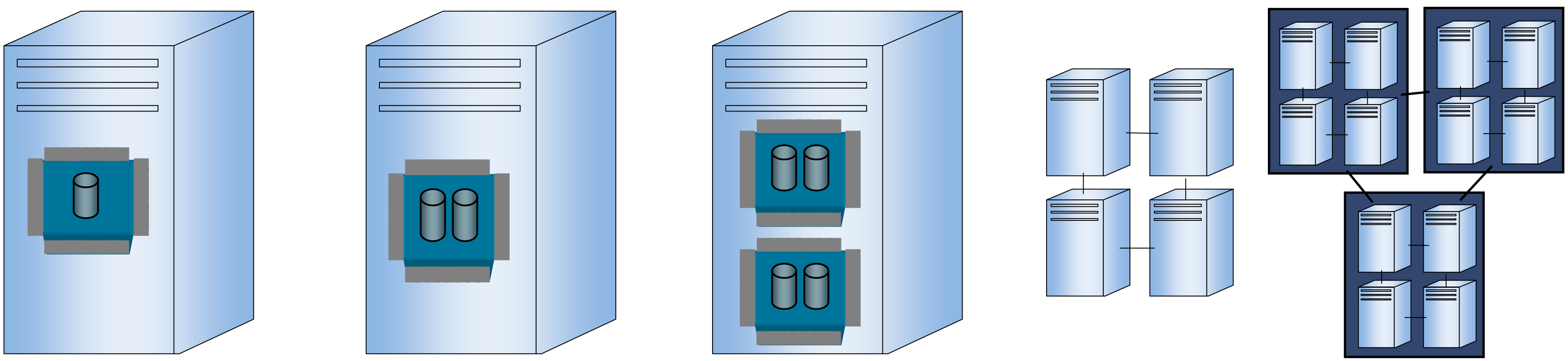




MathWorks
Aerospace and Defense Conference '07

Agenda

- An important trend impacting the Engineering workflow
- Task parallel applications
- Data parallel applications

Market trend: from single processor to grids



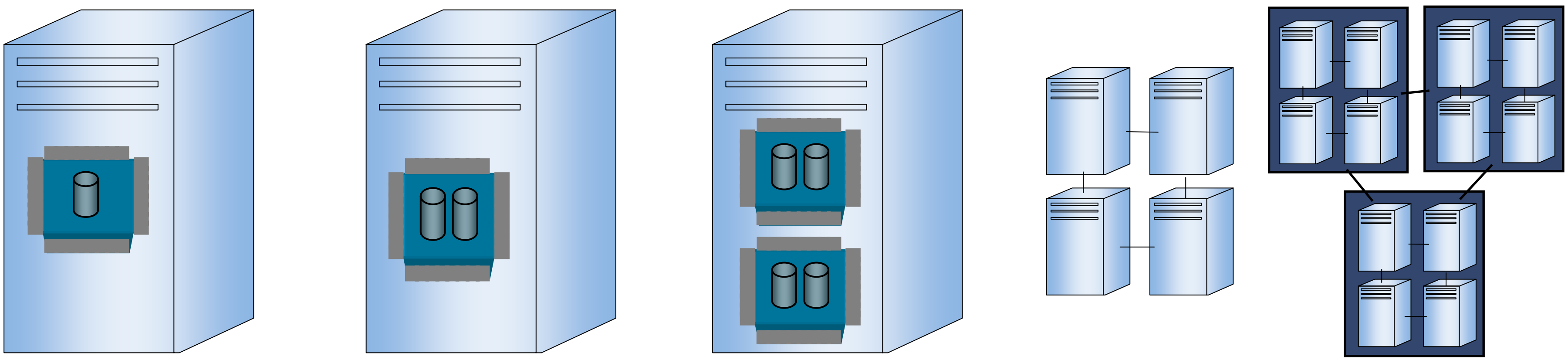
Single processor | Multi-core | Multiprocessor | Clusters | Grids

Why is this important?

- Technical computing
  - Modeling and analysis involves numerous runs
    - Monte carlo or similar applications very common
    - Complexity of algorithms causes longer execution times
  - Data sets are increasing in size

- Model-based design
  - Simulation is done prior to real-world implementation
    - Many scenarios tested
    - Optimal solutions can be found earlier
  - Simulations are growing in complexity and size
    - Simulation time increases
    - Rsim and other targets are only part of the solution

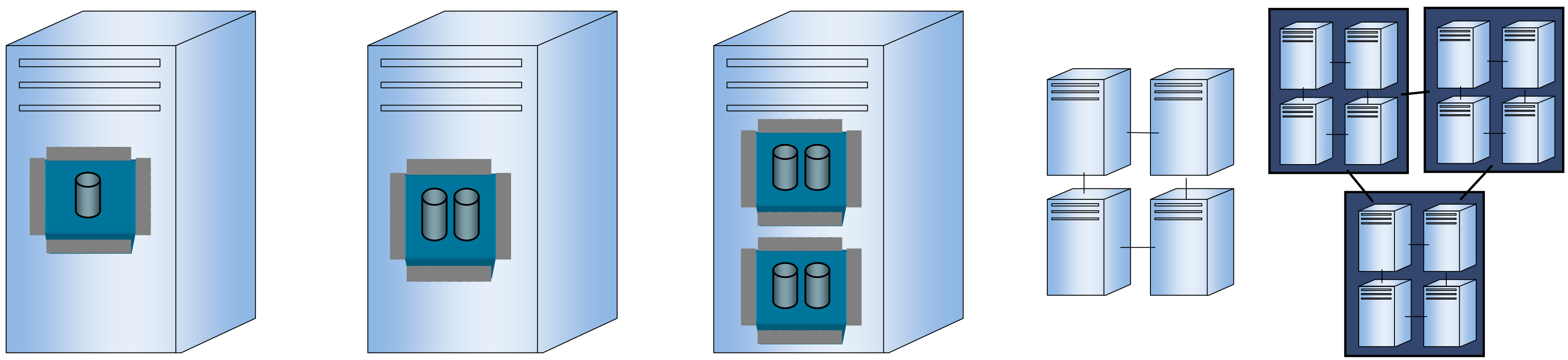
MATLAB addresses the market trend



Single processor | Multi-core | Multiprocessor | Clusters | Grids

MATLAB
in R2006b

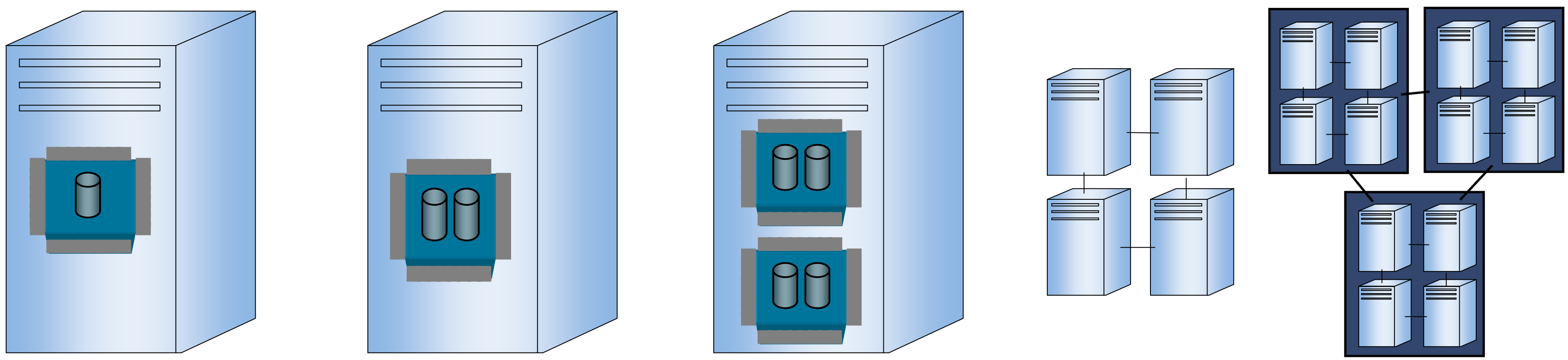
MATLAB addresses the market trend



Single processor | **Multi-core** | **Multiprocessor** | **Clusters** | **Grids**

MATLAB
in R2007a

MATLAB to Distributed Computing



Single processor | **Multi-core** | **Multiprocessor** | **Clusters** | **Grids**

Distributed Computing Toolbox
in R2007a

MATLAB Distributed
Computing Engine

What is happening because of this trend?

Interactive programming - re-use of existing applications

Distributed/parallel engineering software tools becoming available

MPI

OpenMP

ScaLAPACK

Schedulers and Operating Systems providing foundation services

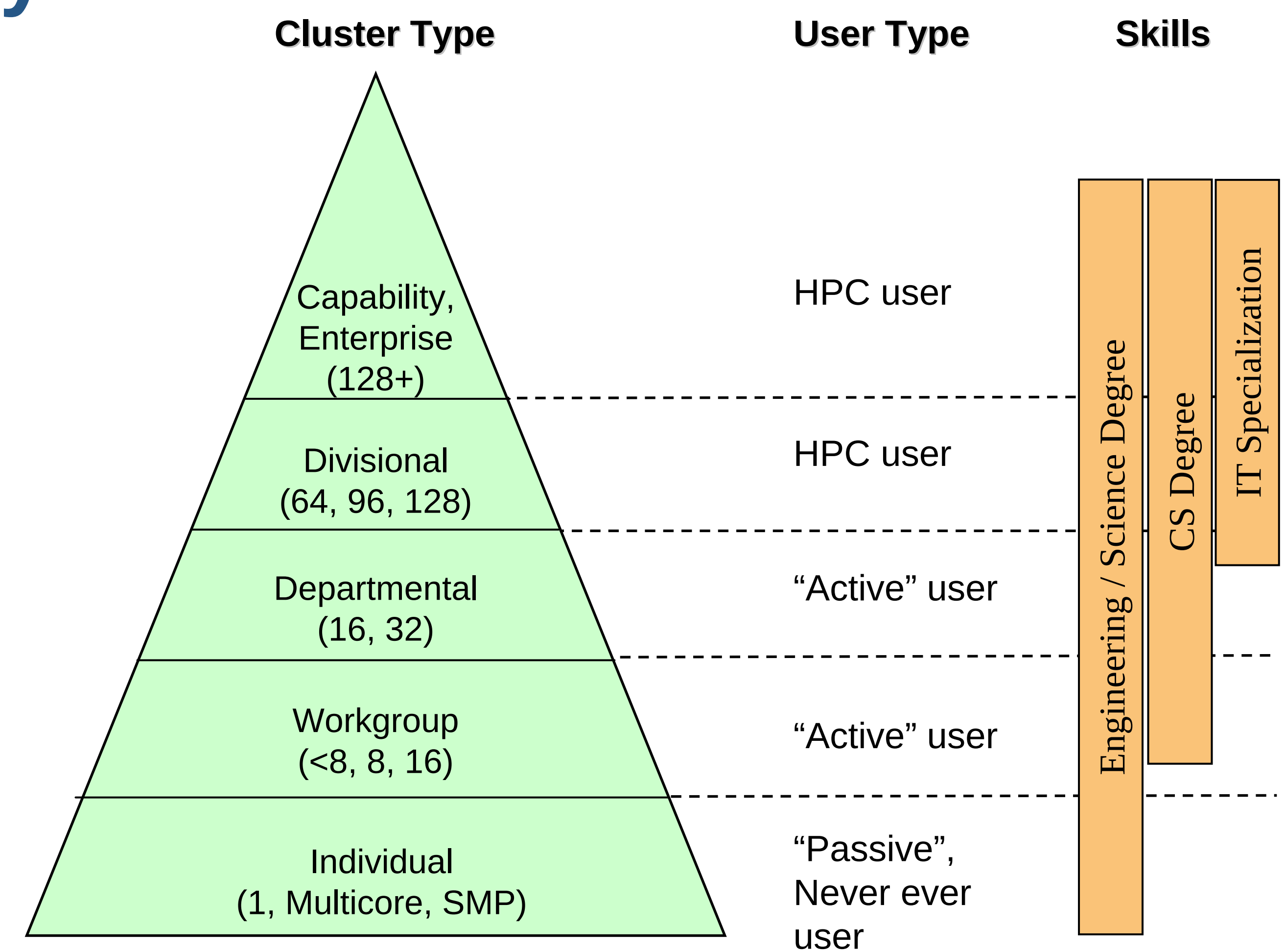
Microsoft Windows Compute Cluster Server 2003

Low-cost hardware

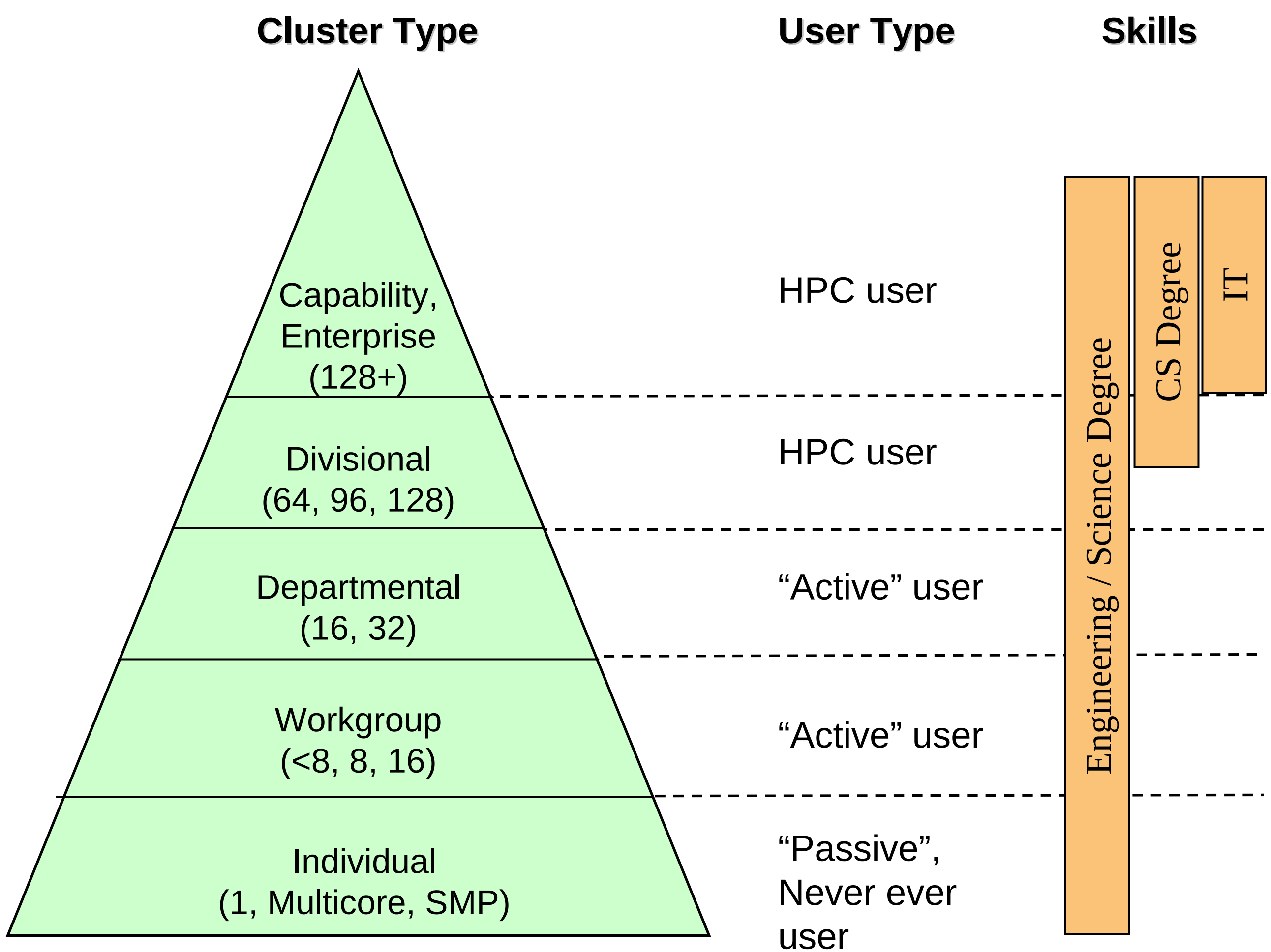
4xShuttle

$4,000

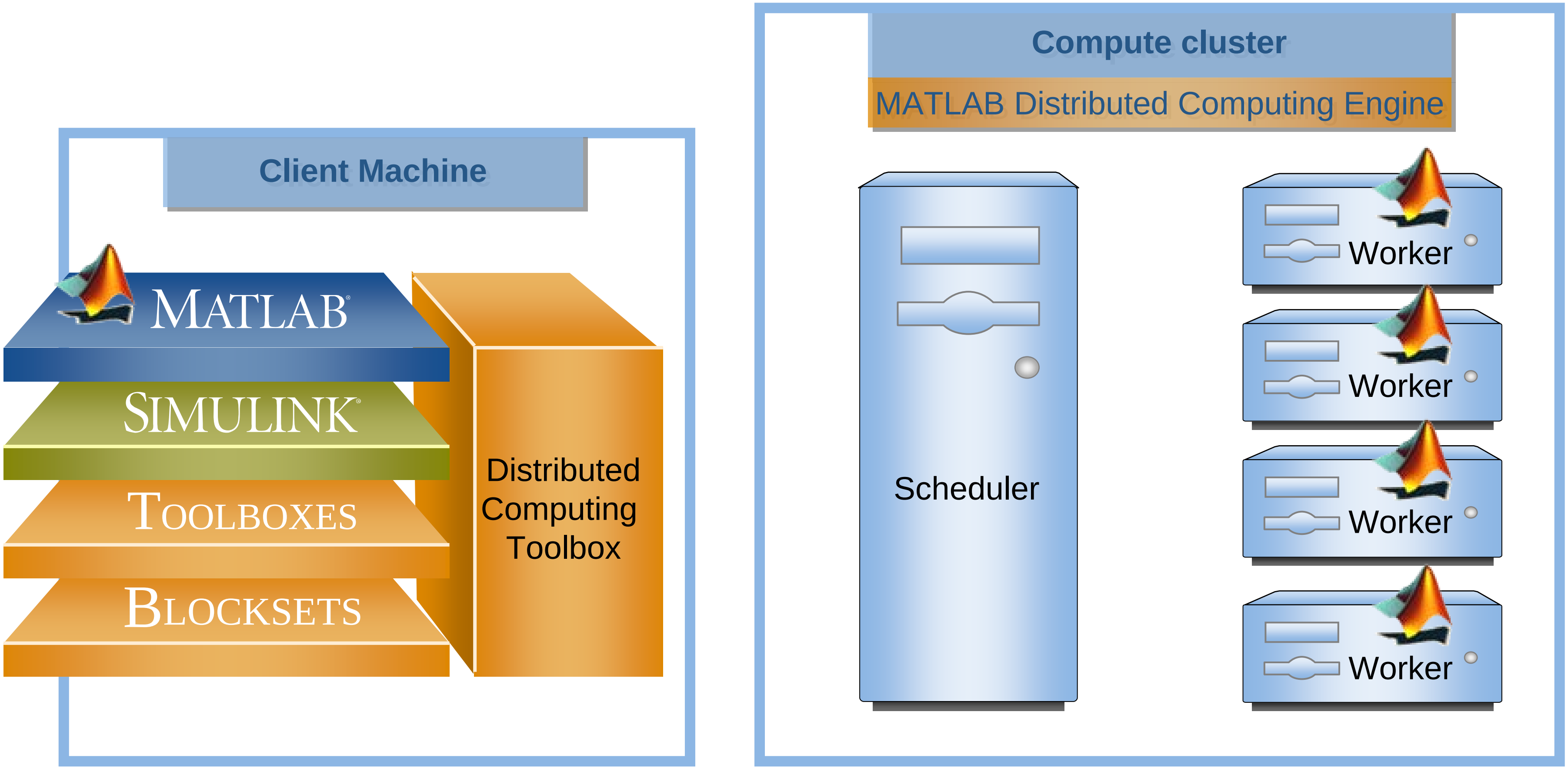
Skills required for distributed computing today



Skills that would be preferred



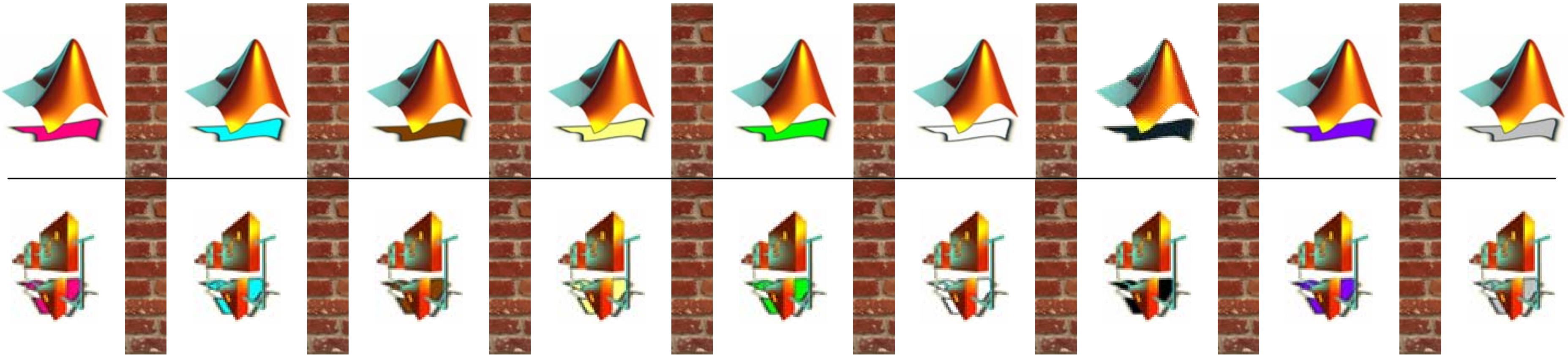
Distributed computing solution



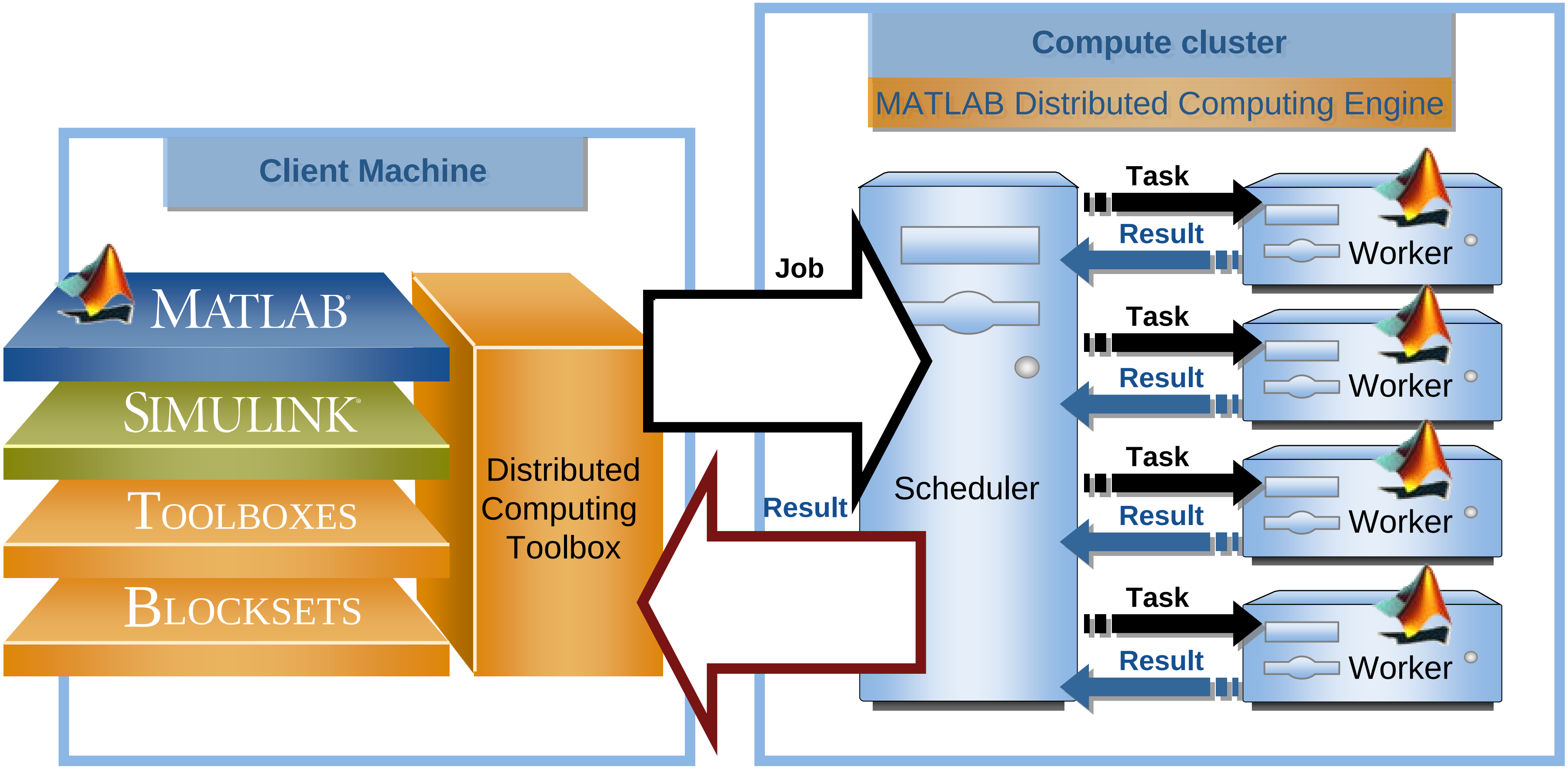
Agenda

- An important trend impacting the Engineering workflow
- Task parallel applications
- Data parallel applications

Multiple independent problems

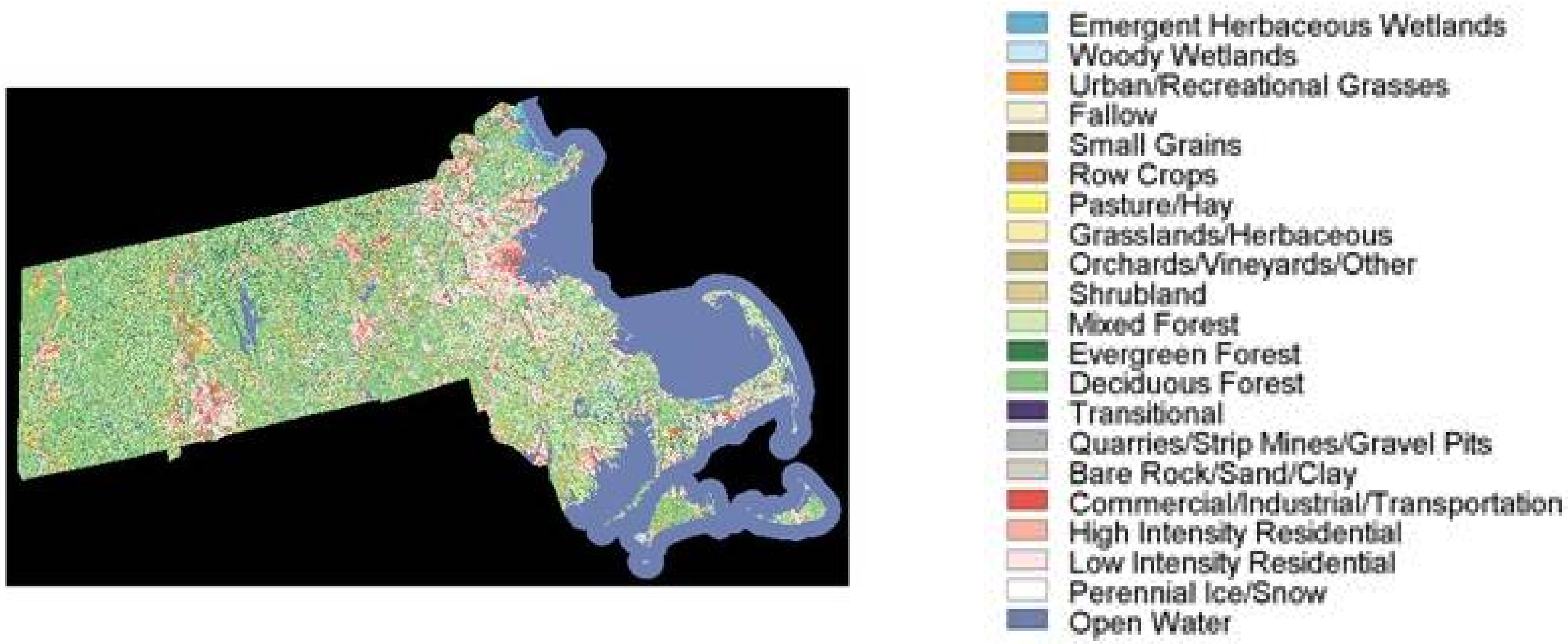


Task parallel applications



Example: Land Classification

- National Land Cover Dataset (NLCD) from U.S. Geological Survey – 30GB
- “Where are wetlands, forests etc concentrated?”
- “How does the distribution compare with other datasets?”



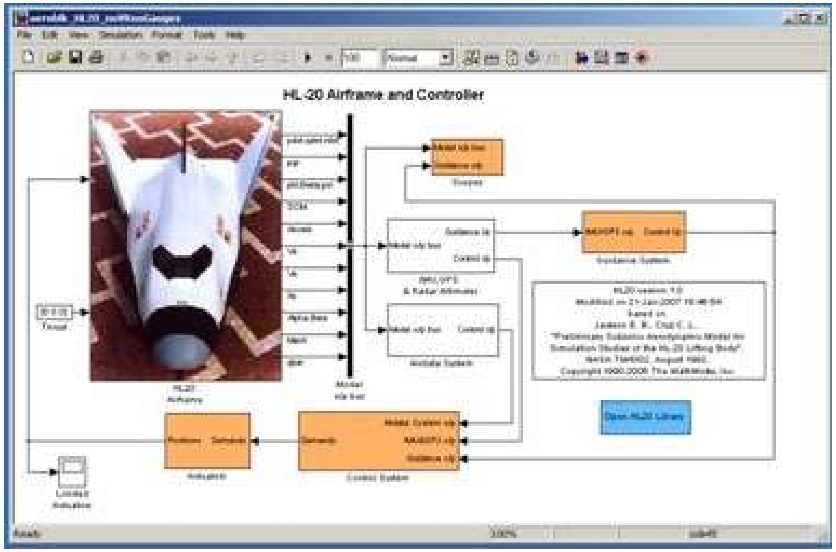
From sequential to distributed: MATLAB

```
function results = main(var1, var2)







nSims = 1000;
out = cell(1, nSims);

for ii = 1:nSims
    out{ii} = myFunction(ii, var1, var2);
end







results = postprocessing(out);
```

From sequential to distributed: MATLAB

```matlab
function results = main(var1, var2)

jm = findResource('scheduler', 'type', 'jobmanager');

job = createJob(jm, ...
    'FileDependencies', 'myFunction.m', ...
    'PathDependencies', {'\\myPath\myFolder\data'});

nSims = 1000;
out = cell(1, nSims);

for ii = 1:nSims
    createTask(job, @myFunction, 1, {ii, var1, var2});
end






results = postprocessing(out);

```

From sequential to distributed: Simulink



1. Divide the Monte Carlo simulations such that each processor executes a full Simulink simulation or RSIM target.

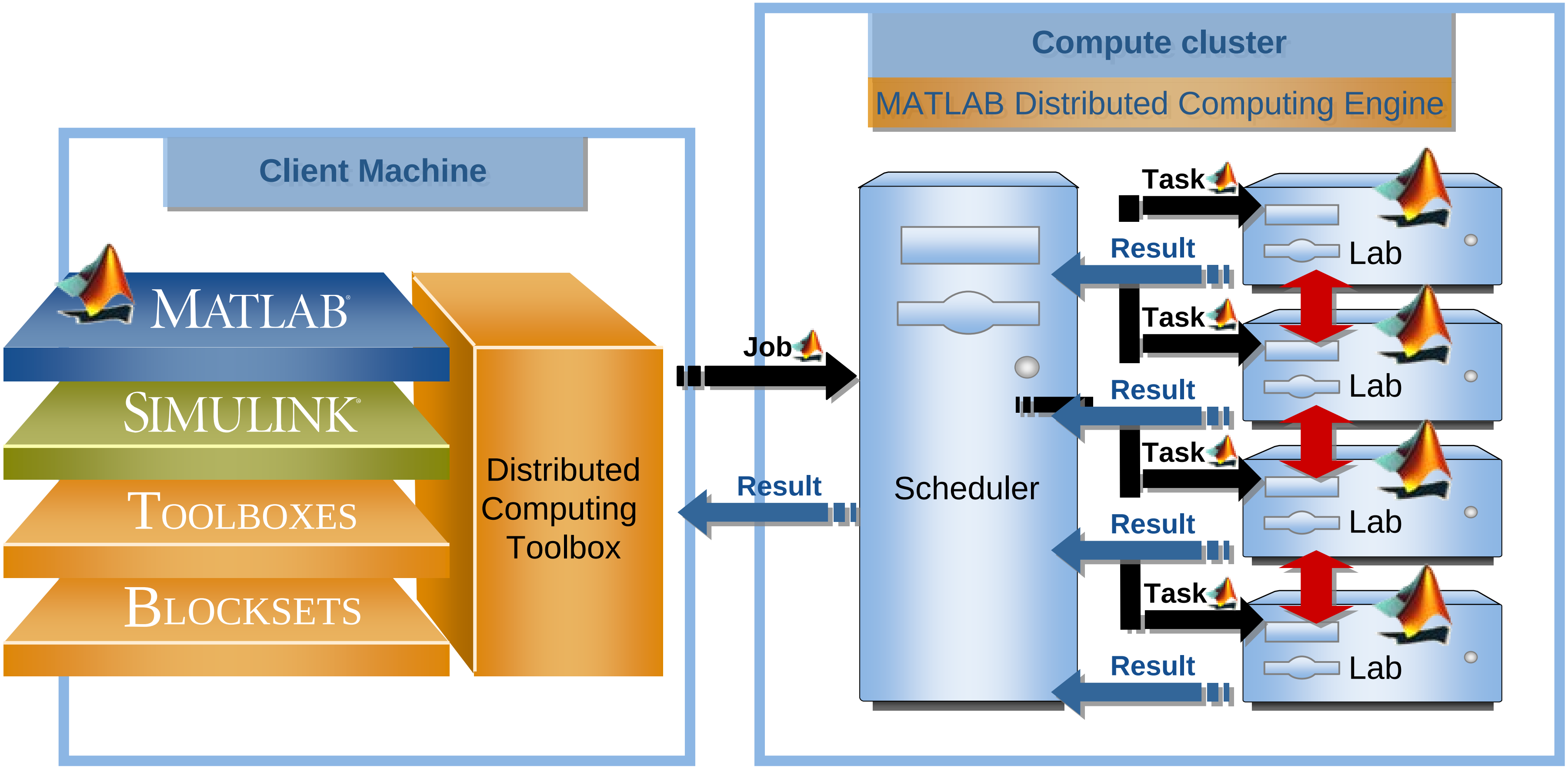
   Eg., one simulation per altitude

2. Create a Task Function that uses MATLAB commands to call the Simulink model you want to execute

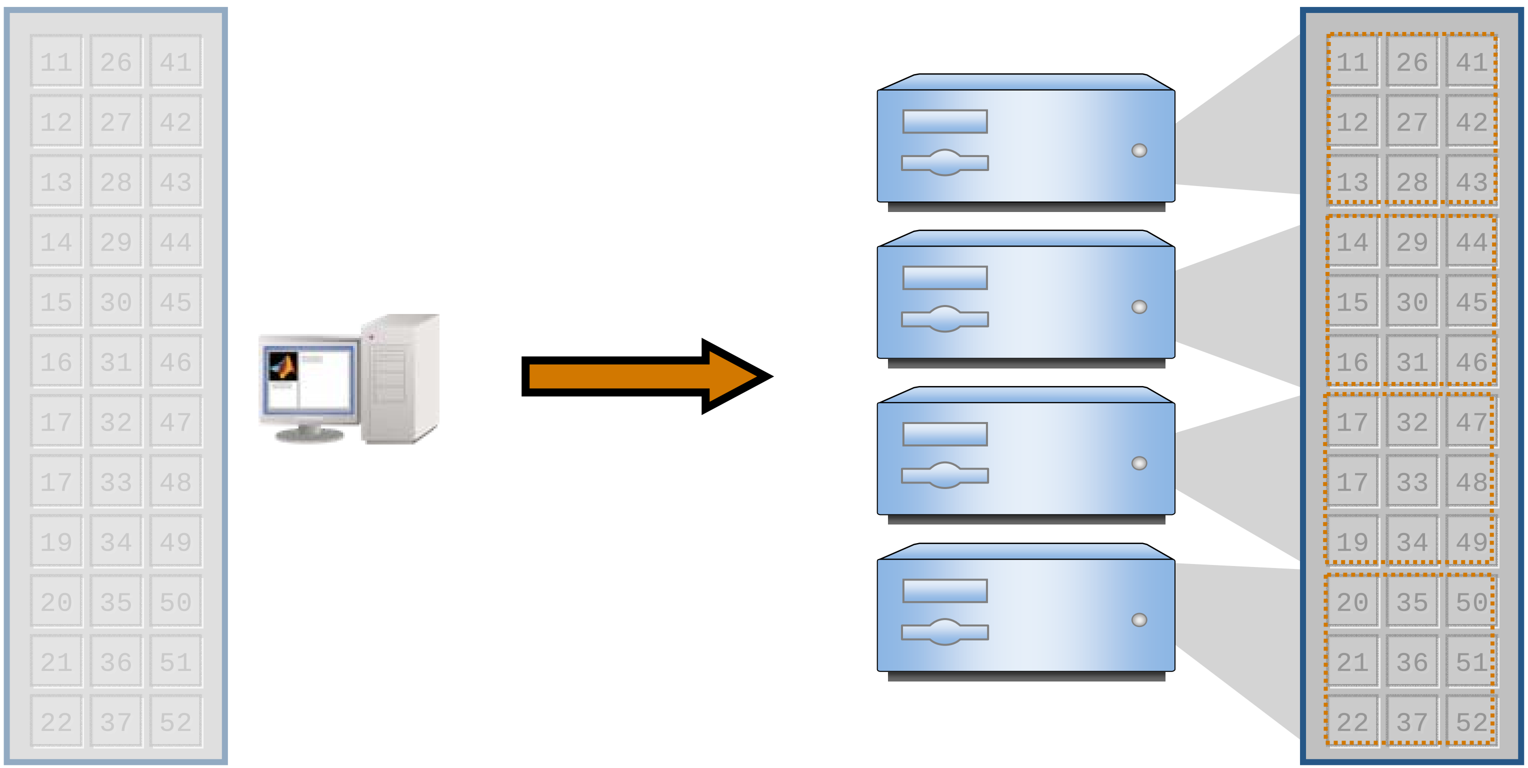
Agenda

- A little history and context setting
- Task parallel solutions
- Data parallel solutions

Data parallel applications
(interactive and batch)



Large Memory Requirements



Transposing a Distributed Matrix

Using FORTRAN and MPI

Using MATLAB and MPI

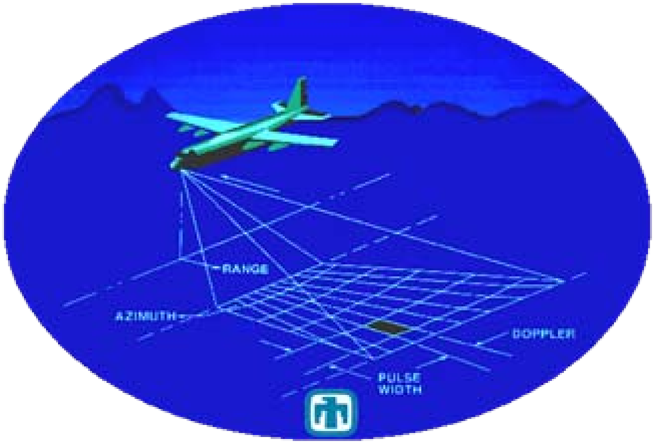
Using Distributed Arrays

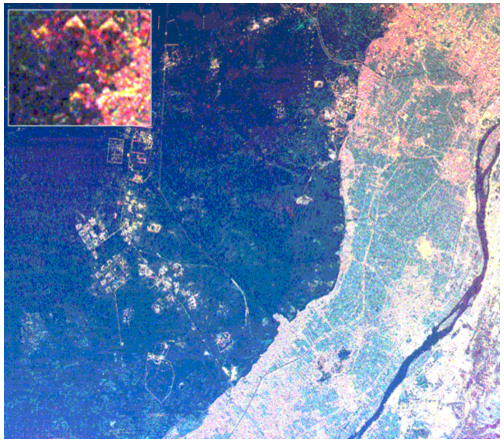
P>> E = D’

Example:

Image Formation Algorithms: Synthetic Aperture Radar (SAR)

- Description
  - SAR is a sophisticated method of post-processing radar data
  - Size and processing requirements demand lots of memory
- Approach
  - Processing SAR images involves interdependent 2-D operations
  - Distribute image across the cluster





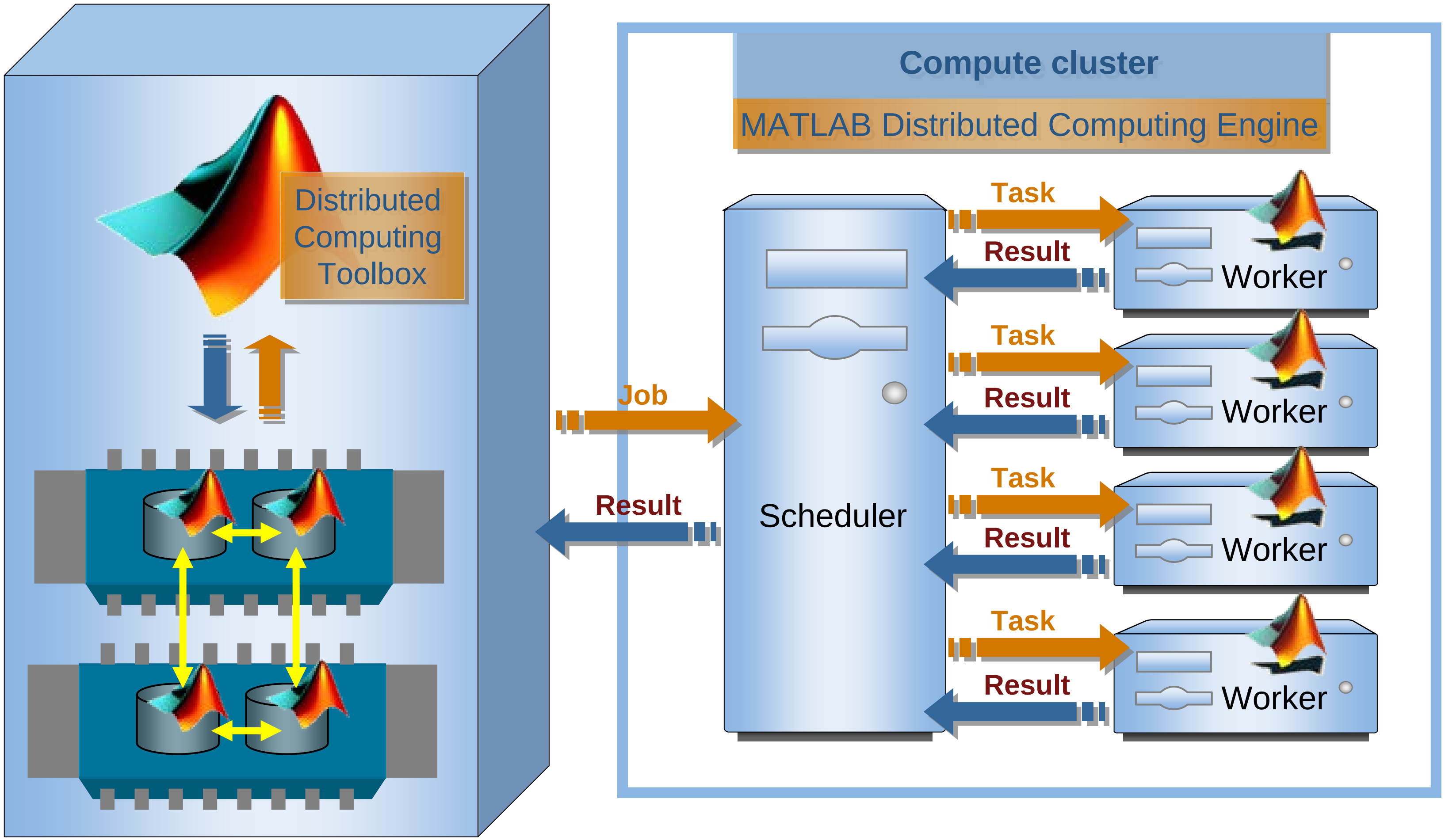
From sequential to distributed: MATLAB

```
function ImageOut = cztproc_single

%Read in SAR image data
load sarimage.mat;
[I,N] = size(fftImage);

im_dist = distribute(fftImage,1);

tic;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% azimuth processing
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
nfft = power(2,nextpow2(2*N-1));
w = exp(-j*2*pi/N);
kk = ( (-N+1):N-1 ).';
kk2 = (kk .^ 2) ./ 2;
ww = w .^ (kk2);   % <----- Chirp filter is 1./ww
nn = (0:(N-1))';
aa = ww(N+nn);

for i = 1:I,
    % Perform azimuth CZT
    x  = im_dist(i,:);
    y = ( x.' ) .* aa;

    fy = fft(  y, nfft );
    fv = fft( 1 ./ ww(1:(2*N-1)), nfft );   % <----- Chirp filter.
    fy = fy .* fv;
    g  = ifft( fy );

    g = [g( N:(2*N-1), : ) .* ww( N:(2*N-1) )].'; %#ok<NBRAK>

    im_dist(i,:) = g;
end
```

From sequential to distributed: MATLAB

```
function ImageOut = cztproc_single

%Read in SAR image data
load sarimage.mat;
[I,N] = size(fftImage);

 im_dist = distribute(fftImage,1);

tic;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% azimuth processing
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
nfft = power(2,nextpow2(2*N-1));
w = exp(-j*2*pi/N);
kk = ( (-N+1):N-1 ).';
kk2 = (kk .^ 2) ./ 2;
ww = w .^ (kk2);   % <----- Chirp filter is 1./ww
nn = (0:(N-1))';
aa = ww(N+nn);

 parfor i = 1:I,
    % Perform azimuth CZT
    x  = im_dist(i,:);
    y = ( x.' ) .* aa;

    fy = fft(  y, nfft );
    fv = fft( 1 ./ ww(1:(2*N-1)), nfft );   % <----- Chirp filter.
    fy = fy .* fv;
    g  = ifft( fy );

    g = [g( N:(2*N-1), : ) .* ww( N:(2*N-1) )].'; %#ok<NBRAK>

    im_dist(i,:) = g;
end
```

Distributed Computing in the Engineering Workflow



Why is this important?

- Technical computing
  - Modeling and analysis involves numerous runs
    - Monte carlo or similar applications very common
    - Complexity of algorithms causes longer execution times
  - Data sets are increasing in size

- Model-based design
  - Simulation is done prior to real-world implementation
    - Many scenarios tested
    - Optimal solutions can be found earlier
  - Simulations are growing in complexity and size
    - Simulation time increases
    - Rsim and other targets are only part of the solution

Summary

- Hardware trends are impacting everybody
- Understanding and creating distributed applications will be an important skill for anybody in the fields of Computer Science or Technical Computing
- MathWorks provides a market-leading solution for distributed applications
- Demo available in exhibit hall