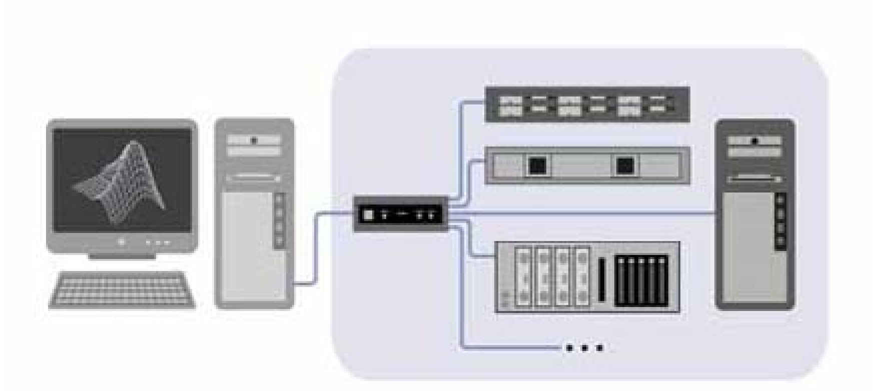
Distributed Computing with MATLAB® and Simulink®

Narfi Stefansson

The MathWorks



The MathWorks
Aerospace & Defense Conference | 2006

Geospatial Application Accelerated with MATLAB and Distributed Computing

The Challenge

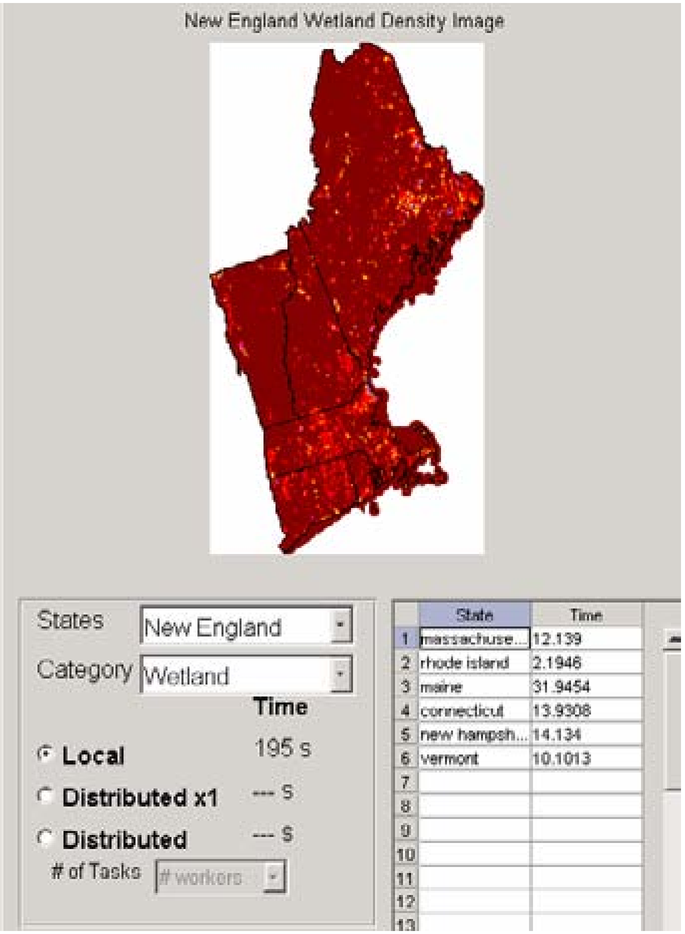
- To compare high-resolution state-by-state land cover maps with lower-resolution U.S. data sets

The Solution

- Use MATLAB®, the Distributed Computing Toolbox, the Image Processing Toolbox, and the Mapping Toolbox to reformulate the original state-by-state land cover data as a single lower resolution U.S. mosaic

The Results

- Computations can run on low-cost computers
- Nearly 4-times speedup on 4 CPUs



New England Wetland Density Image

States: New England

Category: Wetland

Time

● Local 195 s

○ Distributed x1 --- s

○ Distributed --- s

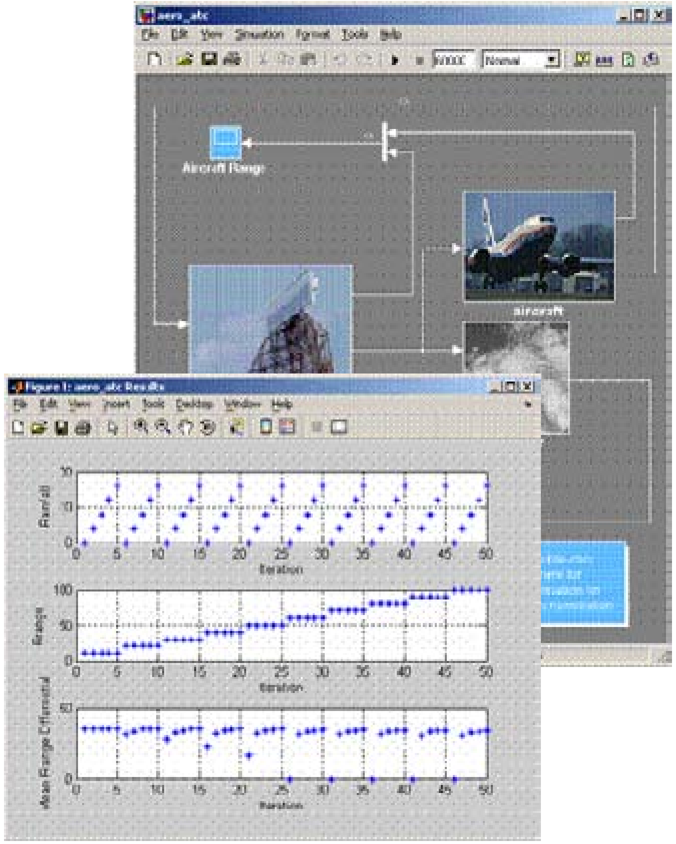
# of Tasks: # workers

	State	Time
1	massachuse...	12.139
2	rhode island	2.1946
3	maine	31.9454
4	connecticut	13.9308
5	new hampsh...	14.134
6	vermont	10.1013
7		
8		
9		
10		
11		
12		
13		

http://www.mathworks.com/company/newsletters/news_notes/jan06/distrib.html

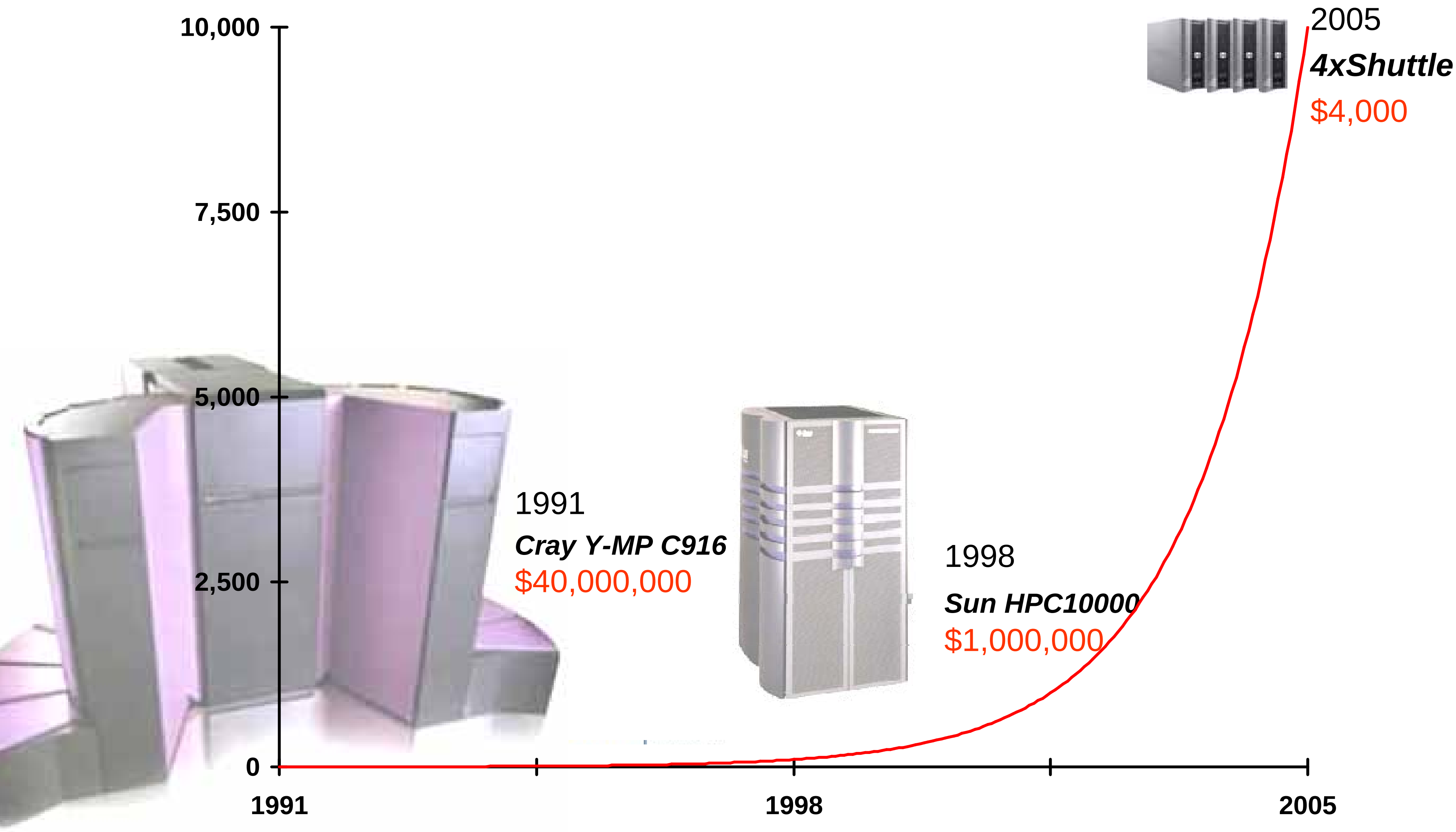
Today’s Agenda

- Introduction
- Key features
- Licensing
- Simulink
- Questions and answers


aero_atc
File Edit View Simulation Format Tools Help
Normal
Aircraft Range
aircraft
Figure 1: aero_atc Results
File Edit View Insert Tools Desktop Window Help
Rainfall
Iteration
Range
Mean Range Differential
0
5
10
15
20
25
30
35
40
45
50
100

The 10 GFlop Personal Computer

10^4 more power for the money vs. 1991



1991
Cray Y-MP C916
$40,000,000

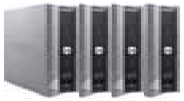
1998
Sun HPC10000
$1,000,000

2005
4xShuttle
$4,000

Standard Operating Systems and Schedulers Now Available

Schedulers
Standard OSs

Low-cost hardware

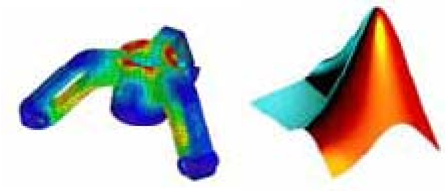


4xShuttle
$4,000

Standard Engineering Software

Apply familiar tools to larger tasks

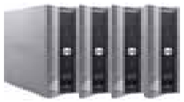
Interactive programming - re-use of existing applications



Distributed/parallel engineering software tools

Schedulers
Standard OSs

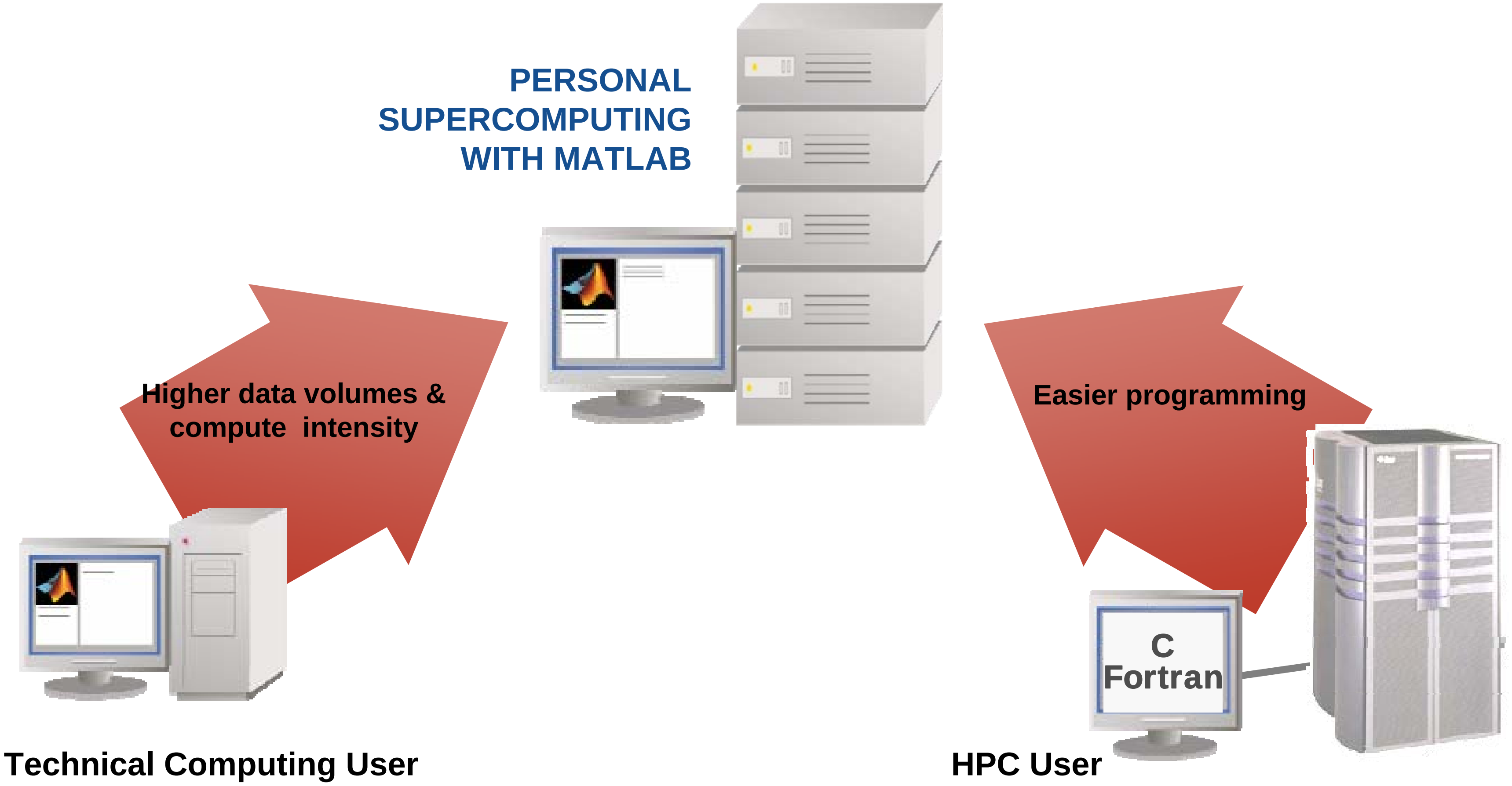
Low-cost hardware



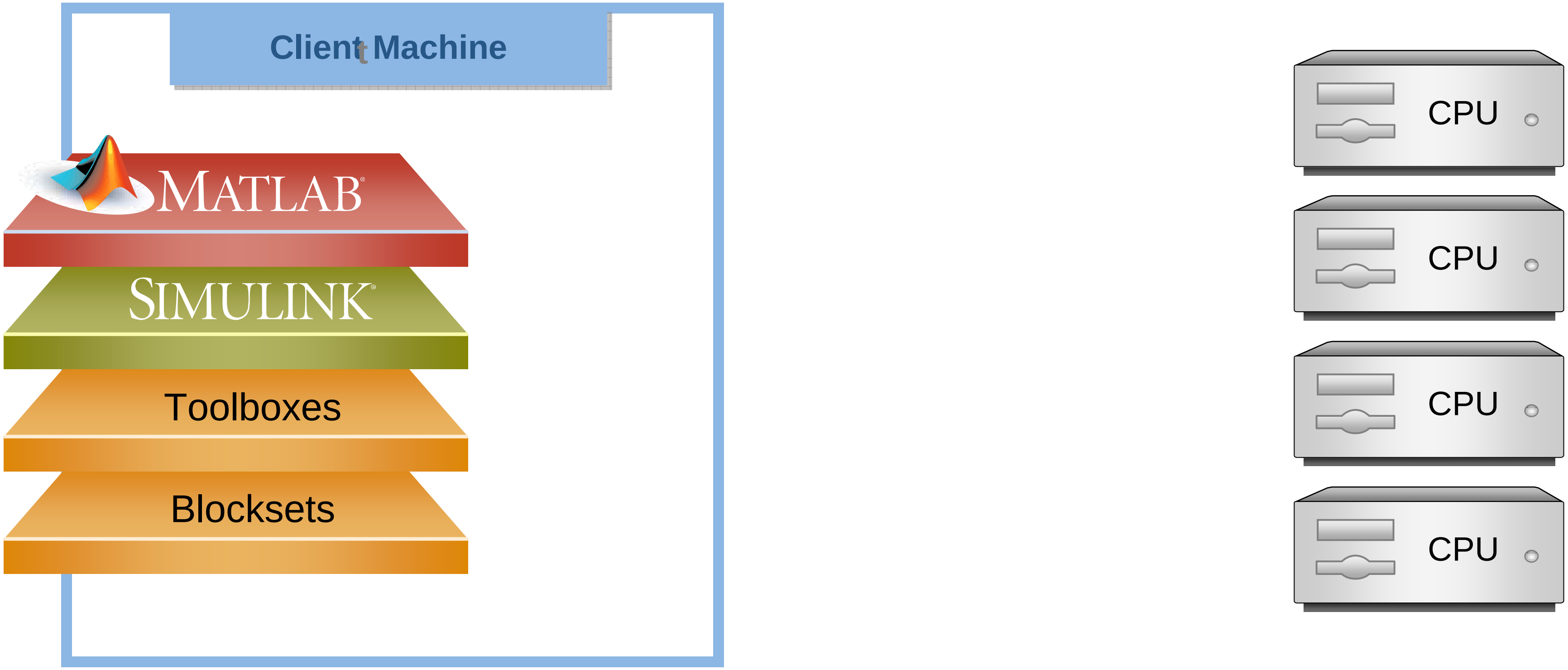
4xShuttle
$4,000

User Requirements

Two user communities


PERSONAL SUPERCOMPUTING WITH MATLAB
Higher data volumes & compute intensity
Easier programming
C
Fortran
Technical Computing User
HPC User

Distributed Computing with MATLAB



Client Machine

MATLAB

SIMULINK

Toolboxes

Blocksets

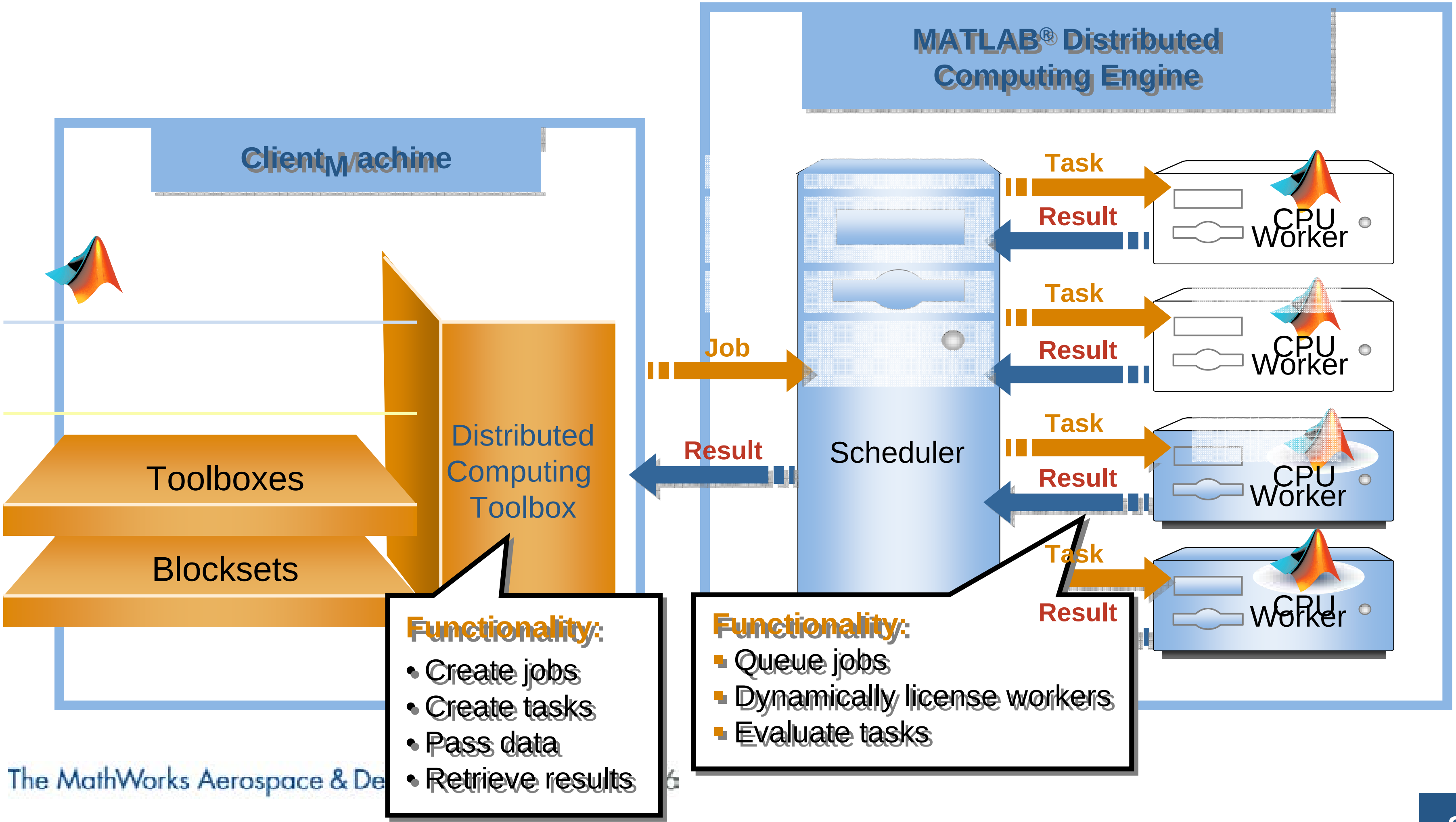
CPU

CPU

CPU

CPU

Distributed Computing with MATLAB



Functionality:

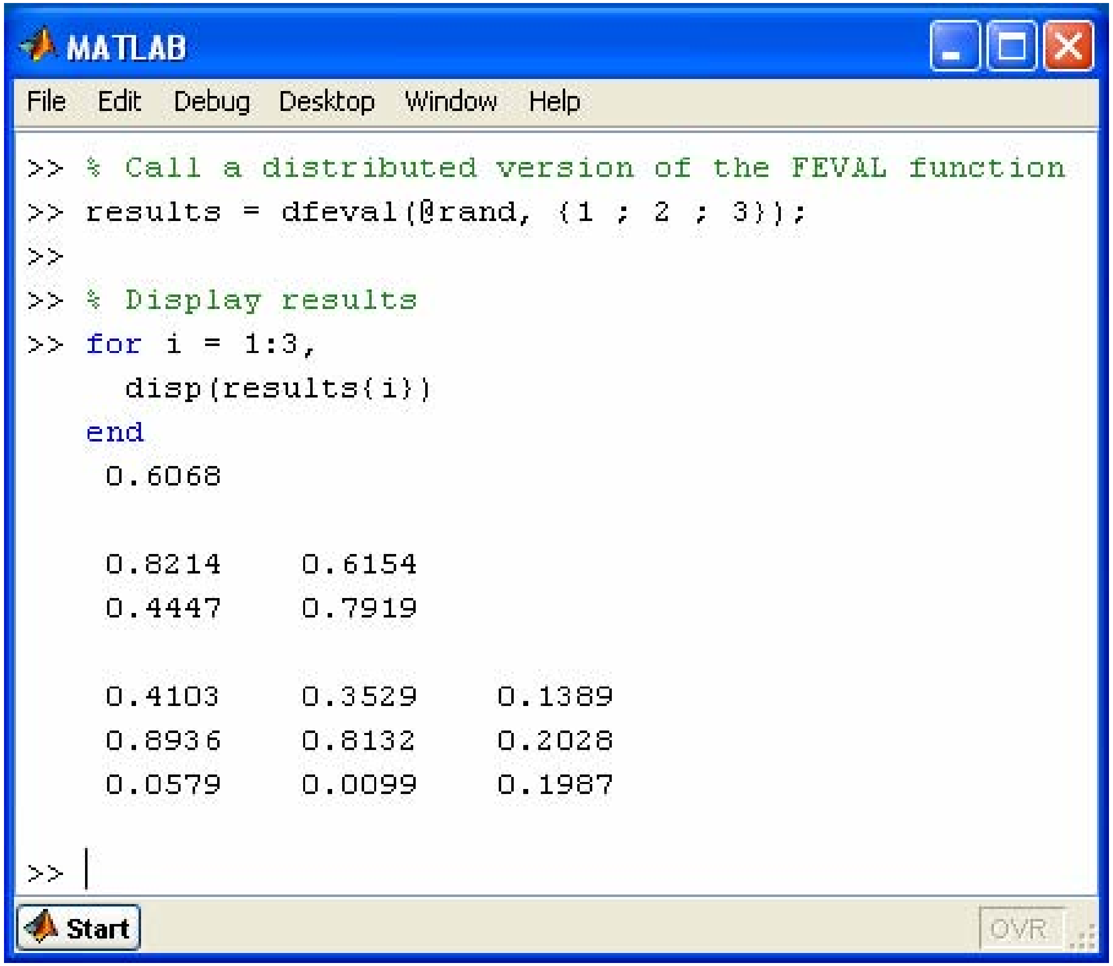
- Create jobs
- Create tasks
- Pass data
- Retrieve results

Functionality:

- Queue jobs
- Dynamically license workers
- Evaluate tasks

Familiar Interface

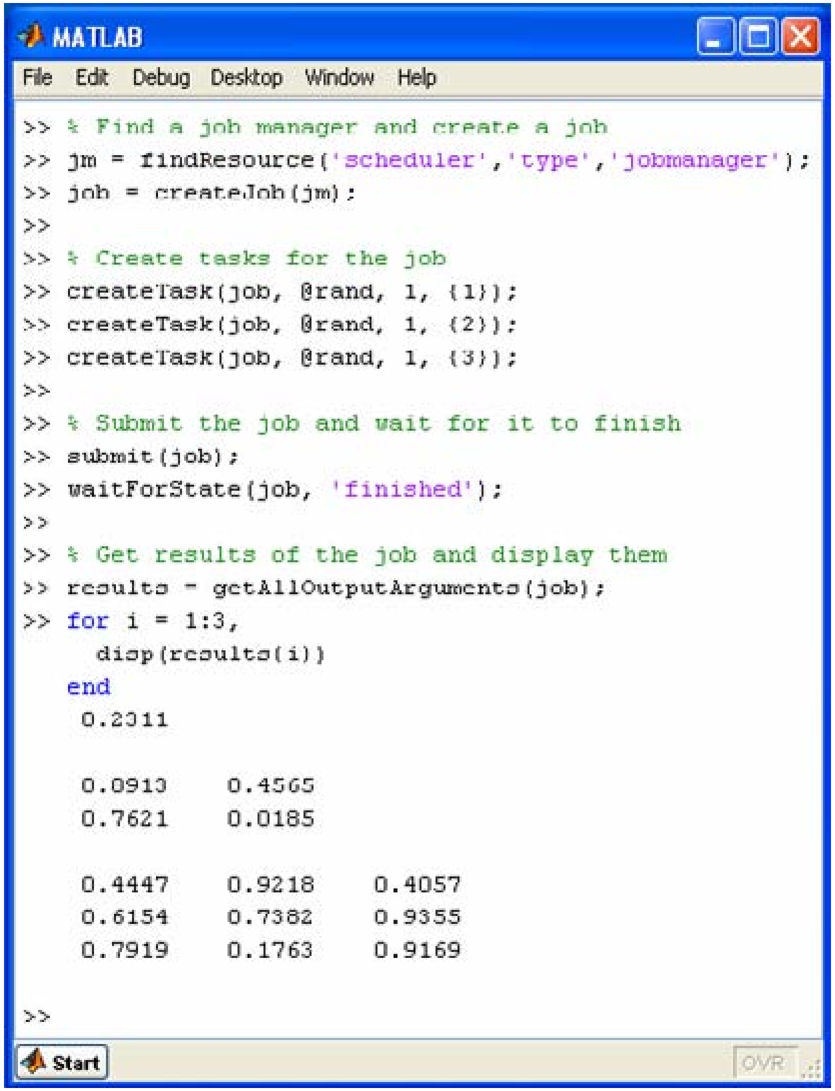
Function-based interface



```
>> % Call a distributed version of the FEVAL function
>> results = dfeval(@rand, {1 ; 2 ; 3});
>>
>> % Display results
>> for i = 1:3,
     disp(results{i})
   end
    0.6068

    0.8214    0.6154
    0.4447    0.7919

    0.4103    0.3529    0.1389
    0.8936    0.8132    0.2028
    0.0579    0.0099    0.1987

>> 
```

Full Control Over Job Handling

Object-based interface

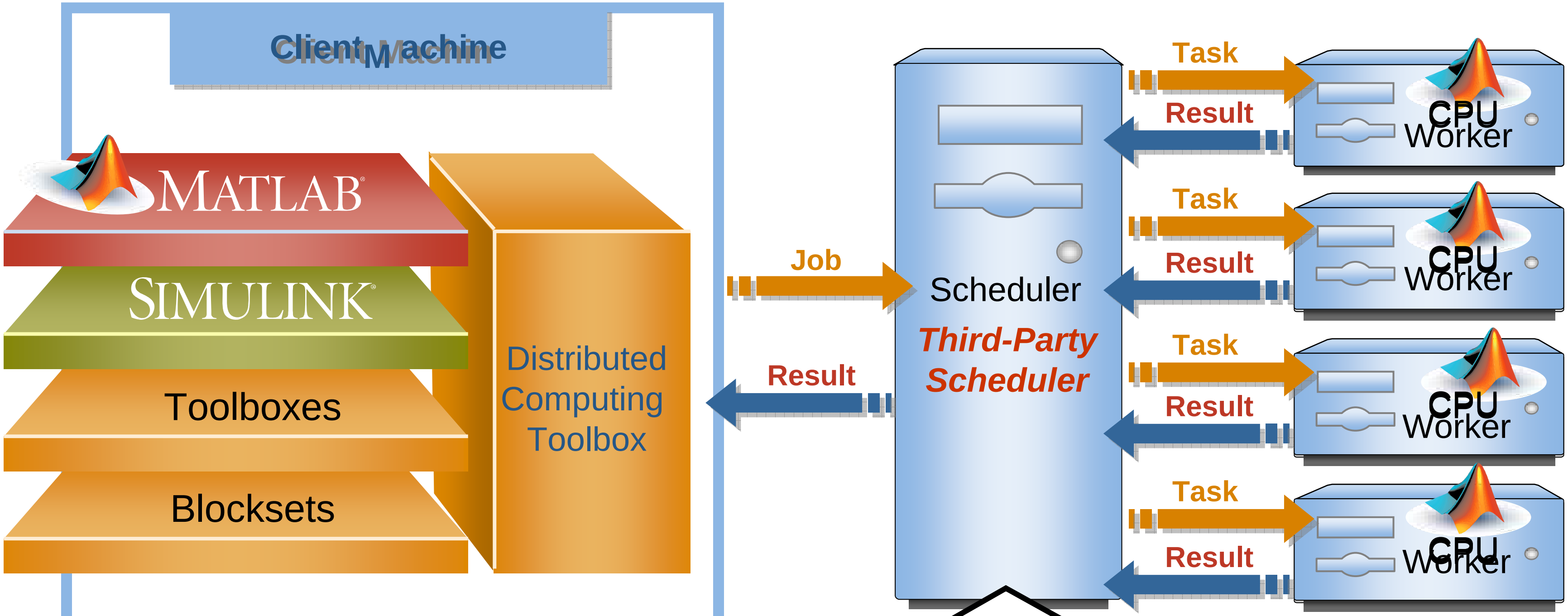


```
>> % Find a job manager and create a job
>> jm = findResource('scheduler','type','jobmanager');
>> job = createJob(jm);
>>
>> % Create tasks for the job
>> createTask(job, @rand, 1, {1});
>> createTask(job, @rand, 1, {2});
>> createTask(job, @rand, 1, {3});
>>
>> % Submit the job and wait for it to finish
>> submit(job);
>> waitForState(job, 'finished');
>>
>> % Get results of the job and display them
>> results = getAllOutputArguments(job);
>> for i = 1:3,
     disp(results{i})
   end
    0.2311

    0.0913    0.4565
    0.7621    0.0185

    0.4447    0.9218    0.4057
    0.6154    0.7382    0.9355
    0.7919    0.1763    0.9169

>>
```

Other Key Features

- Schedulers
- Parallel applications
- Varied modes of interaction
- Hardware

Support for Third-Party Schedulers



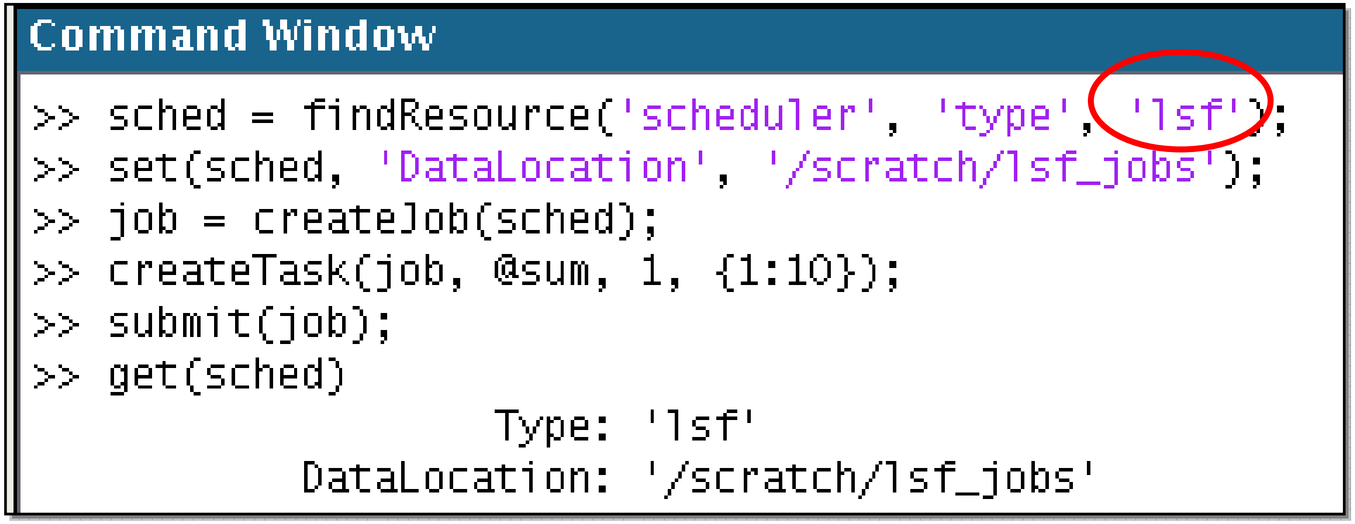
Out-of-the-box support:

- MathWorks built-in scheduler (job manager)
- Platform Computing LSF
- API for integration with additional schedulers - SunGE and PBS examples posted on Web site

Benefits of Scheduler Integration

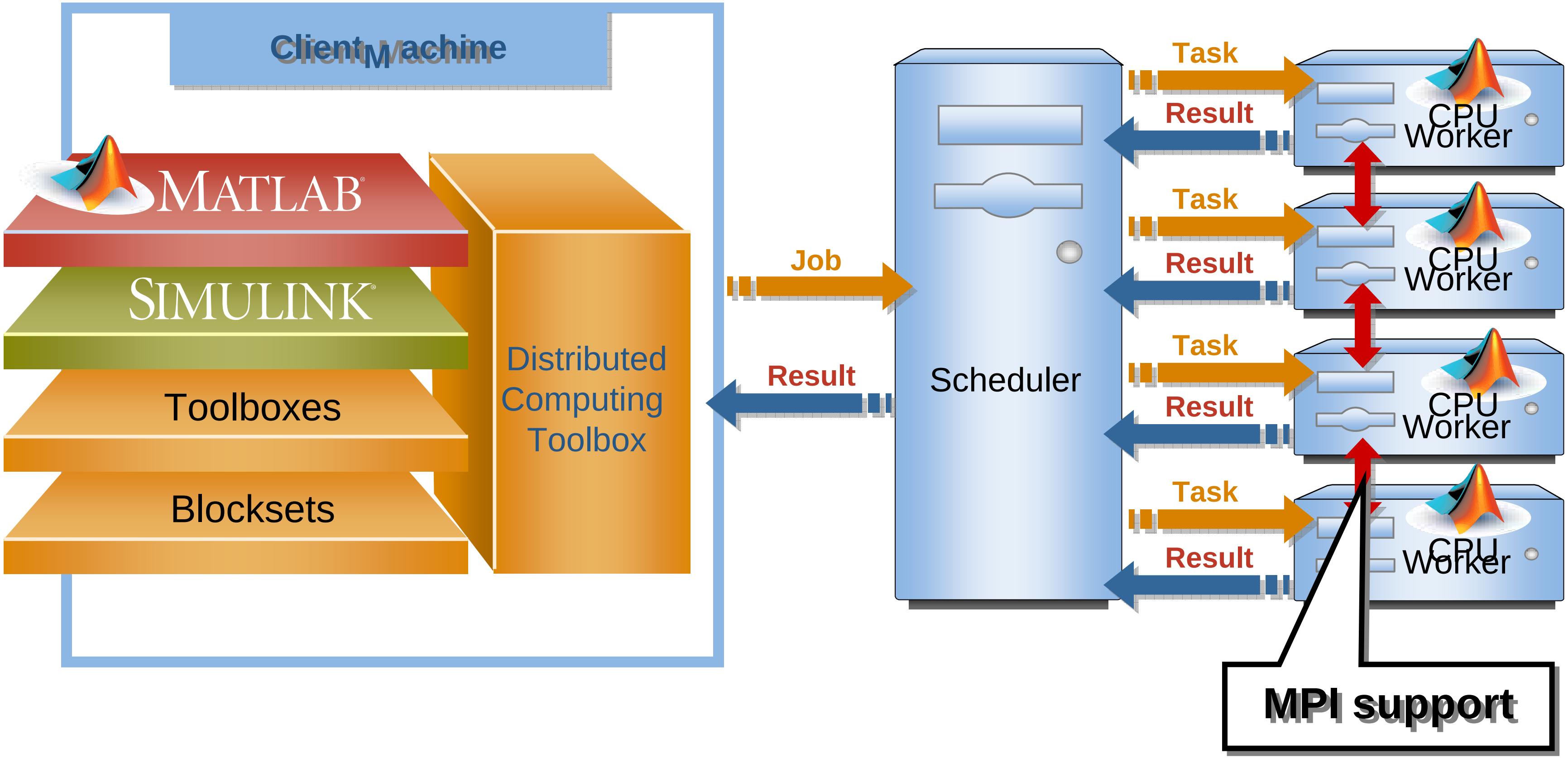
- Take advantage of the scheduler unique capabilities
  - Advanced scheduling
  - Batch workflow support
  - Utilization and performance increase
  - Scalability, reliability, and security
- Run MATLAB and other applications on same cluster
  - Increased throughput
  - Reduced costs of ownership

Simple to Customize for Third-Party Schedulers



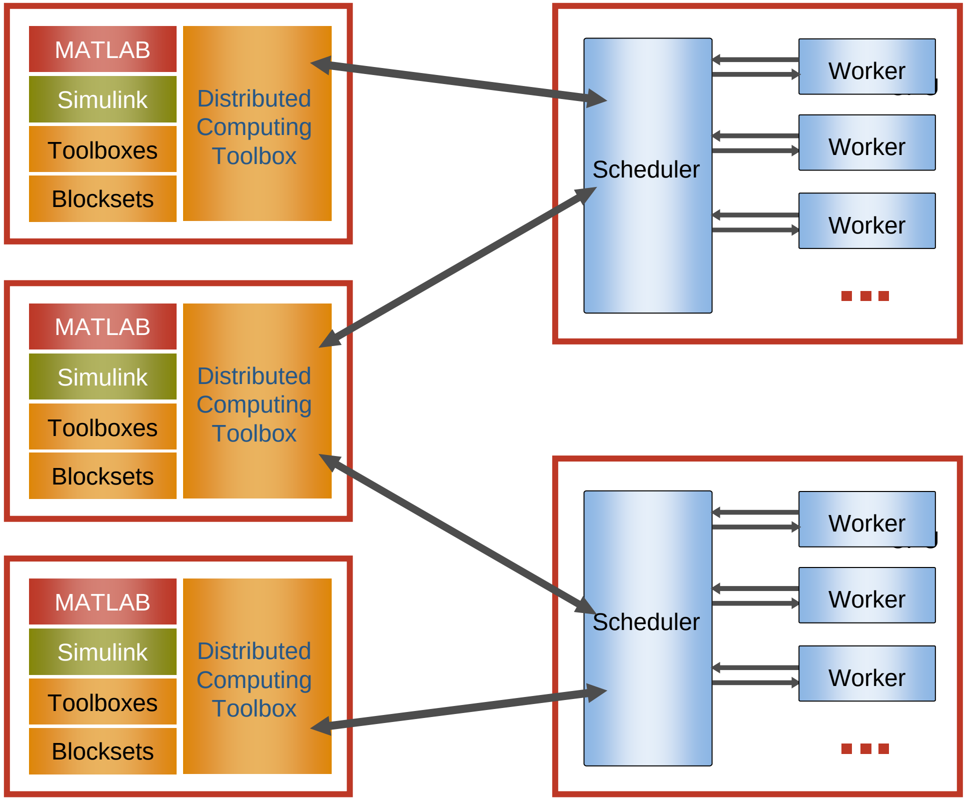
```
>> sched = findResource('scheduler', 'type', 'lsf');
>> set(sched, 'DataLocation', '/scratch/lsf_jobs');
>> job = createJob(sched);
>> createTask(job, @sum, 1, {1:10});
>> submit(job);
>> get(sched)
                    Type: 'lsf'
            DataLocation: '/scratch/lsf_jobs'
```

Support for Parallel Applications



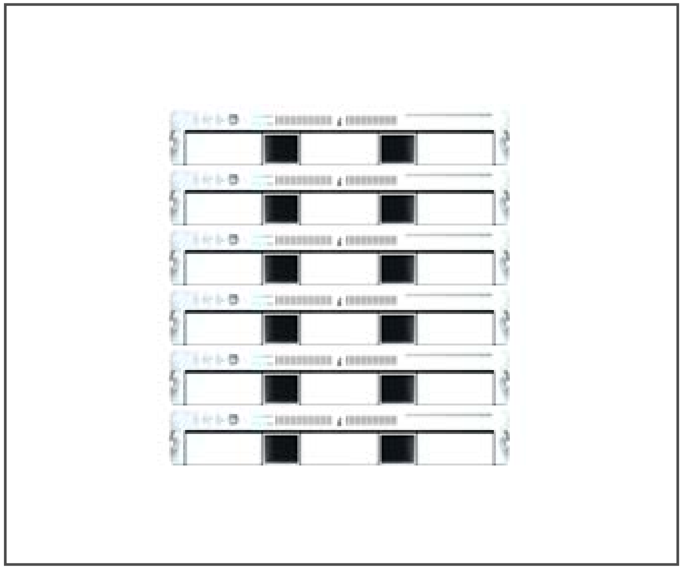
Versatile Modes of Access

Access to single or multiple clusters by single or multiple users (one to many, many to one)

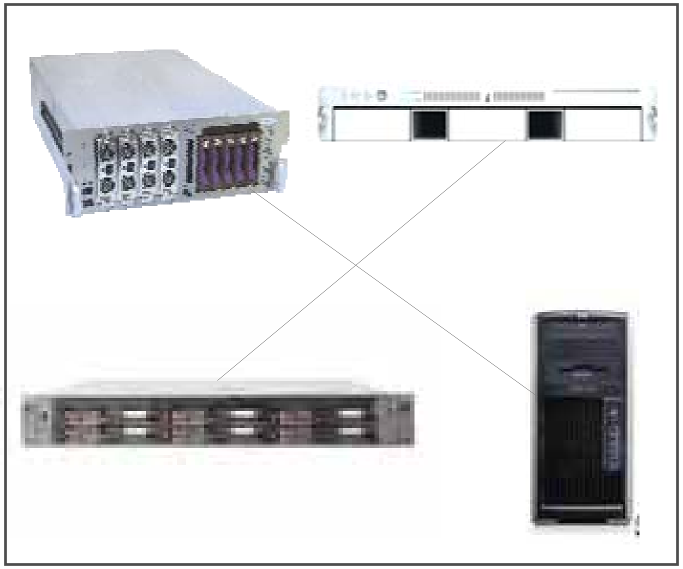


Heterogeneous Platform Support

Support for all platforms that MATLAB supports



Homogeneous cluster

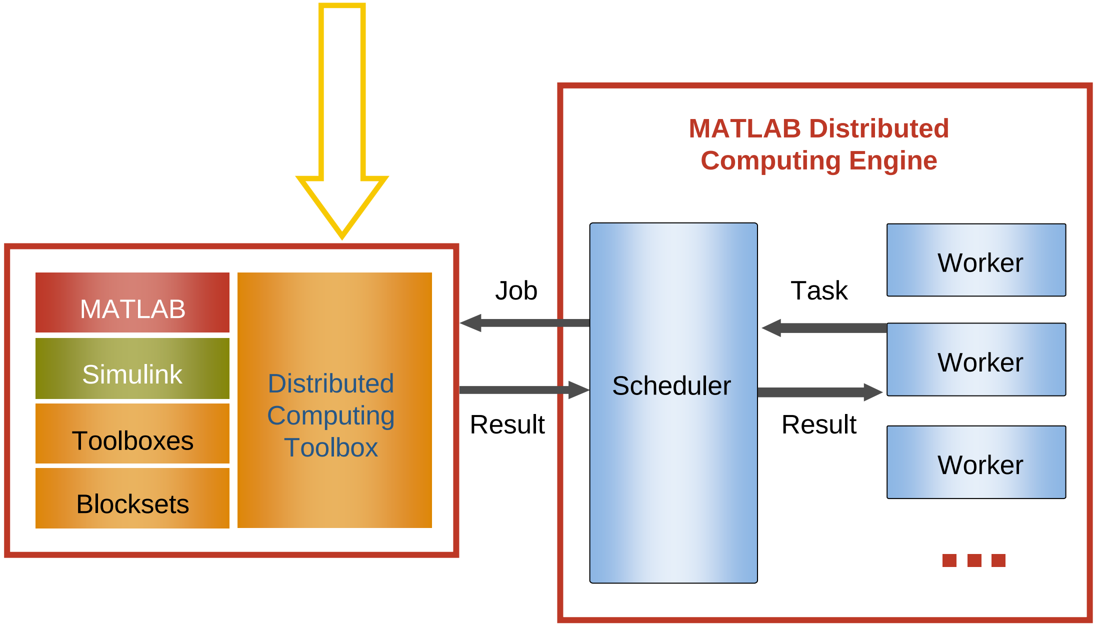


Heterogeneous cluster

Distributed Computing Toolbox

Licensed like any other toolbox

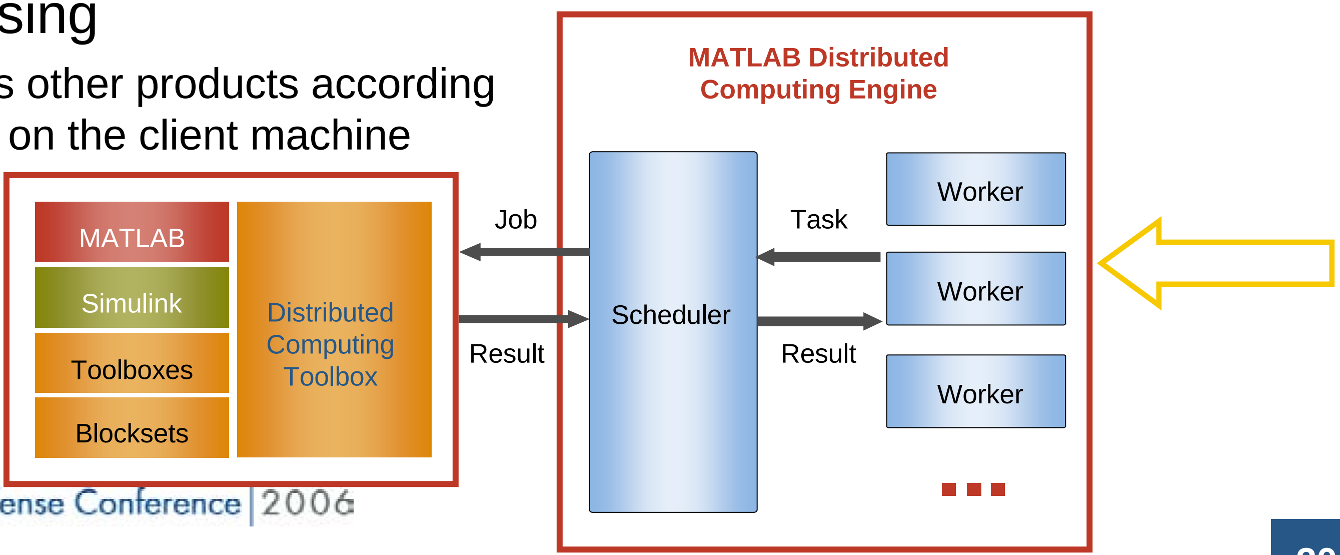
- Individual, concurrent, group
- Requires MATLAB



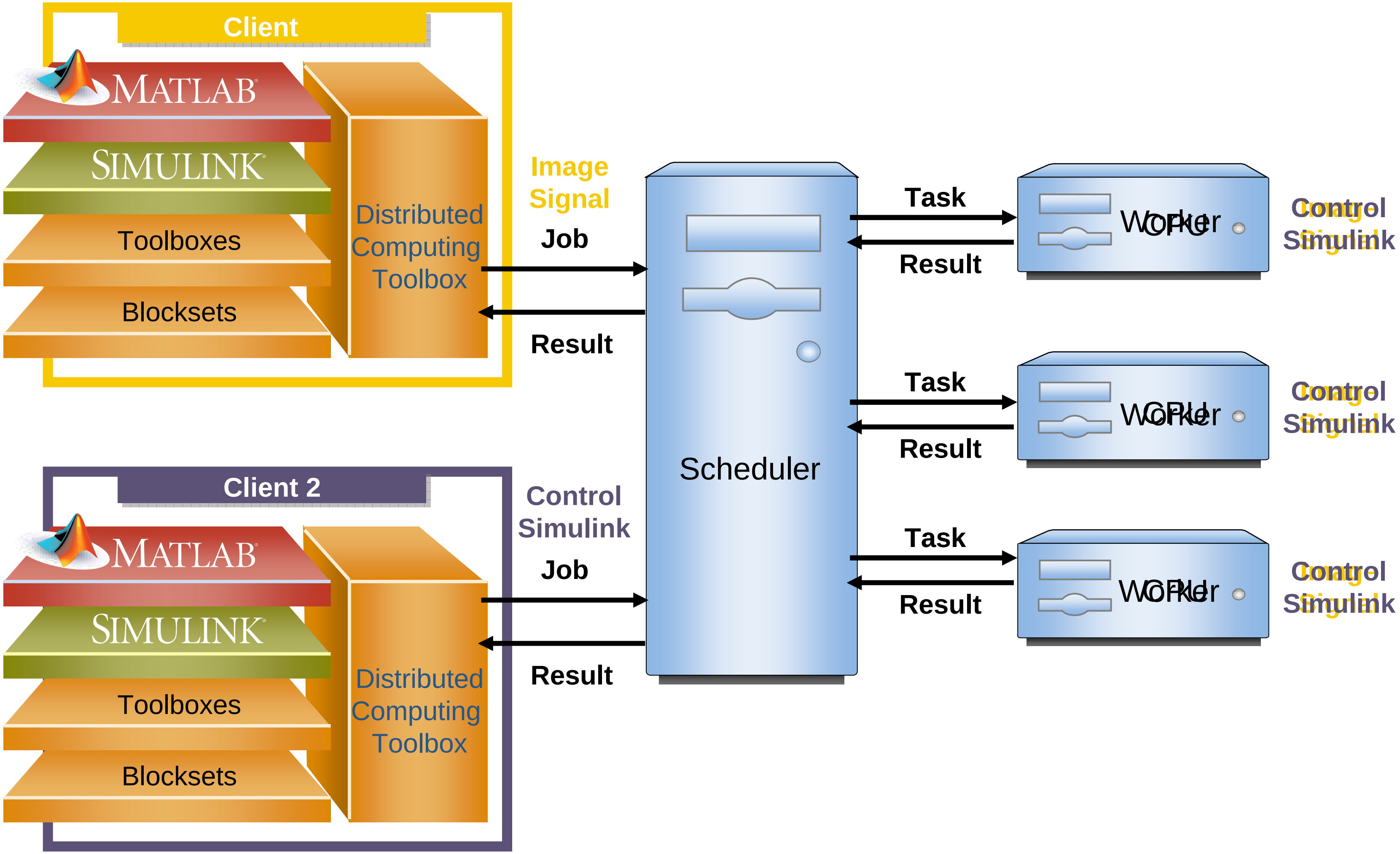
MATLAB Distributed Computing Engine

Cost-efficient licensing

- One key required per worker (to MATLAB session, not a processor)
  - Sold in packs of 8, 16, 32, 64, 96, 128, etc.
  - Scheduler not license managed
- All-product install
  - Code generation and deployment products excluded
  - Simulink and related products do not work with MPI
- Dynamic licensing
  - Engine enables other products according to user license on the client machine



Dynamic Licensing



Research Engineers Advance Design of the International Linear Collider with MathWorks Tools

Queen Mary high-throughput cluster.

The Challenge

To design a control system for ensuring the precise alignment of particle beams in the International Linear Collider

The Solution

Use MATLAB, Simulink, the Distributed Computing Toolbox, and the Instrument Control Toolbox to design, model, and simulate the accelerator and alignment control system

The Results

- Simulation time reduced by an order of magnitude
- Development integrated
- Existing work leveraged

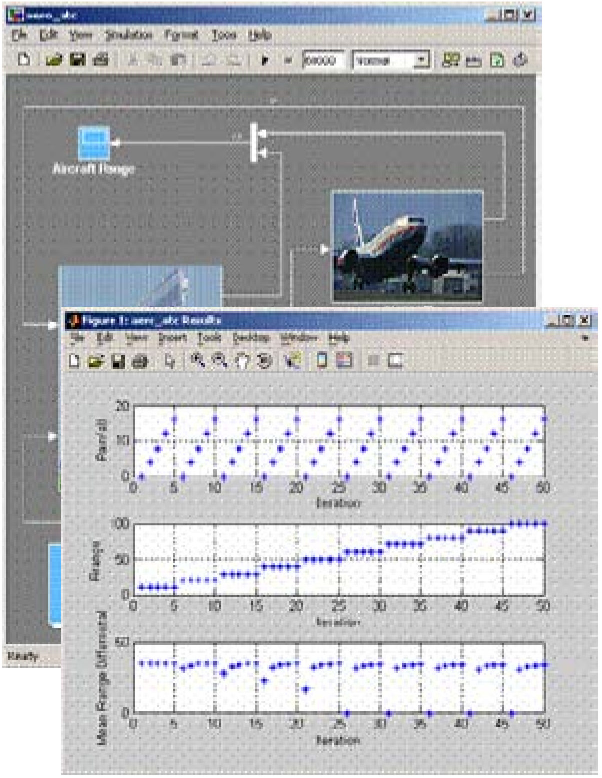
> **"Using the Distributed Computing Toolbox, we simply deployed our simulation on a large group cluster. We saw a linear improvement in speed, and we could run 100 simulations at once. MathWorks tools have enabled us to accomplish work that was once impossible."**
>
> **Dr. Glen White,**
> **Queen Mary, University of London**

Applying Distributed Computing to Simulink

```
for lp = 1:nSims,
  results{lp}=sim(‘model’, …)
end
```

end

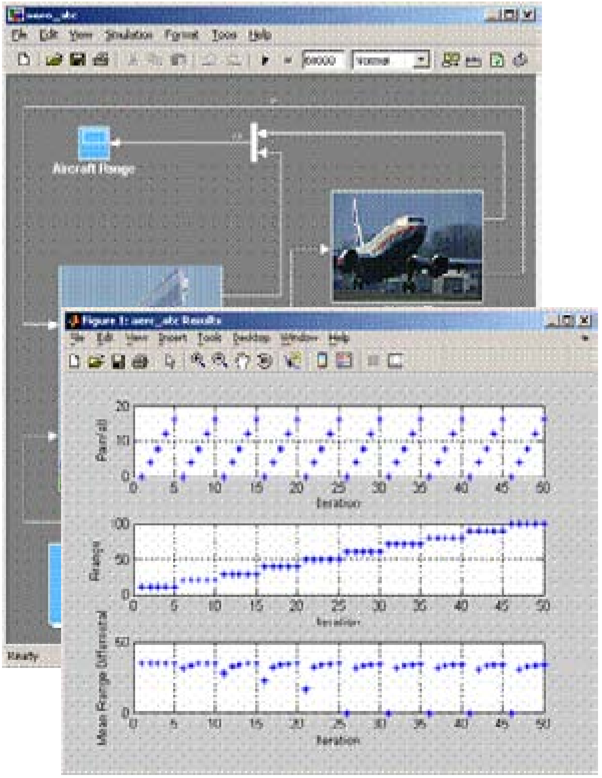
```
results = dfeval(@simunit,{nSims});

function results=simUnit
results=sim(‘model’, …)
```



Applying distributed computing to Simulink

```
for lp = 1:nSims,
  results{lp}=sim(‘model’, …)
end
```

```
sched = findResource(…);
job=createJob(sched);
for lp = 1:nSims,
  createTask(job,@simUnit, …)
end
submit(job)
waitForState(job,’finished’);
results = getAllOutputArguments(job);

function results=simUnit
results=sim(‘model’, …)
```



Distributed Computing Tools Summary

- Increase productivity by reducing the development time of distributed applications
- Improve performance
  - Easy to develop distributed MATLAB applications
  - Third-party scheduler support
  - Distributed and parallel execution
  - Dynamic licensing
- Preview an upcoming release in exhibit hall

Questions?