MathWorks

MATLAB&SIMULINK

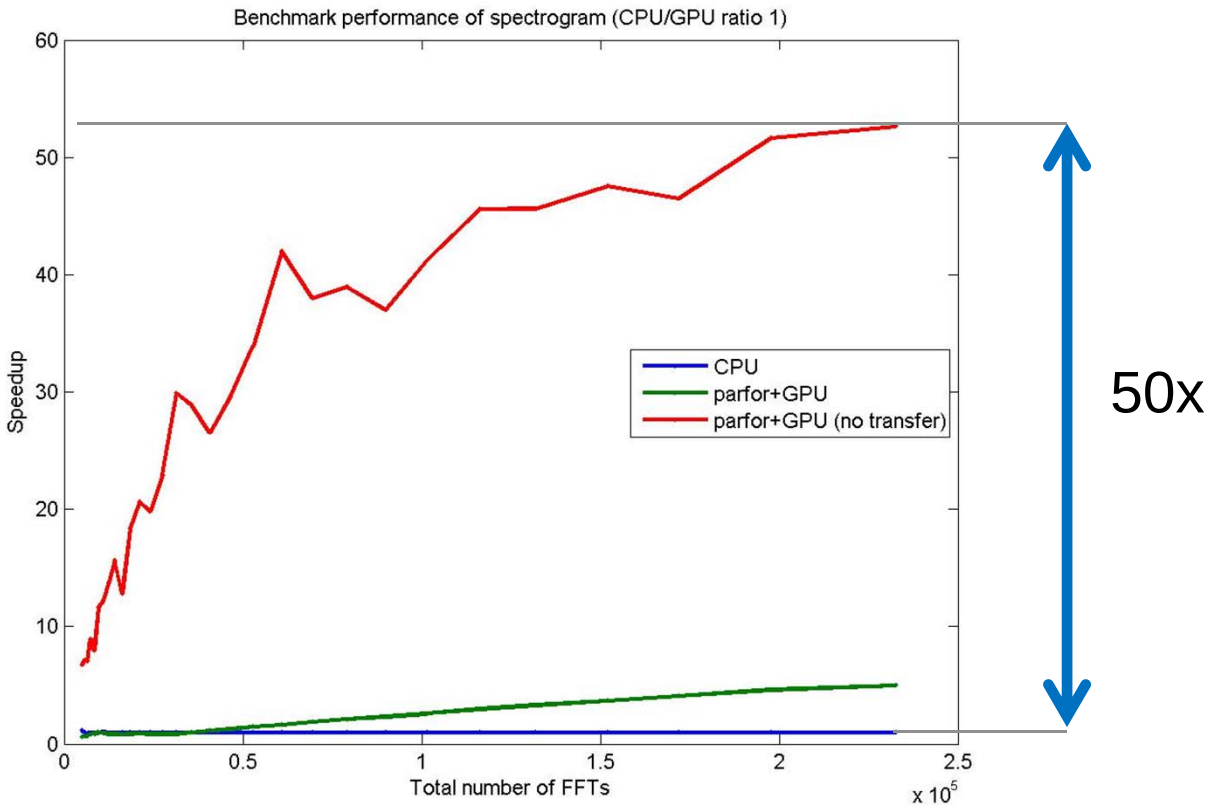
GPU Computing with MATLAB

Loren Dean
Director of Engineering, MATLAB Products
MathWorks



© 2010 The MathWorks, Inc.

Spectrogram shows 50x speedup in a GPU cluster



Agenda

- Background
- Leveraging the desktop
  - Basic GPU capabilities
  - Multiple GPUs on a single machine
- Moving to the cluster
  - Multiple GPUs on multiple machines
- Q&A

How many people are using…

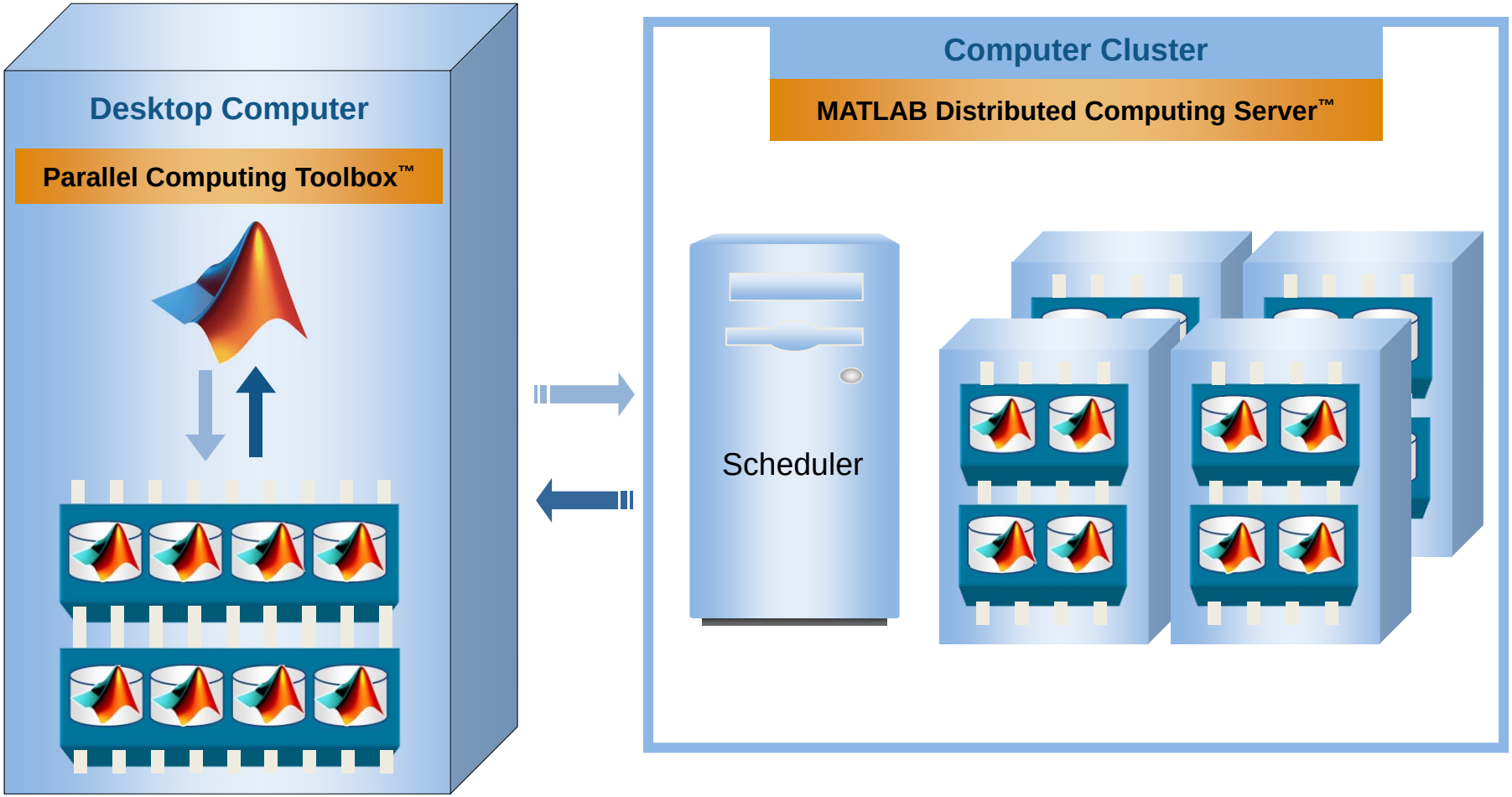
- MATLAB
- MATLAB with GPUs
- Parallel Computing Toolbox
  - R2010b prerelease
- MATLAB Distributed Computing Server

Why GPUs and why now?

- Operations are IEEE Compliant
- Cross-platform support now available
- Single/double performance inline with expectations

Parallel Computing with MATLAB

Tools and Terminology

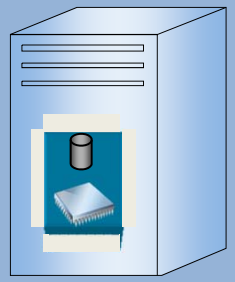


Parallel Capabilities

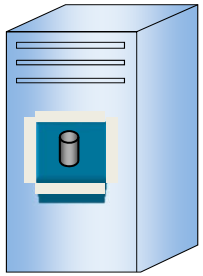


Task Parallel	Data Parallel	Environment
Built-in support with Simulink, toolboxes, and blocksets		`matlabpool` Local workers
`parfor`	distributed array >200 functions	Configurations `batch` MathWorks job manager
`job/task`	`spmd` co-distributed array MPI interface	third-party schedulers `job/task`

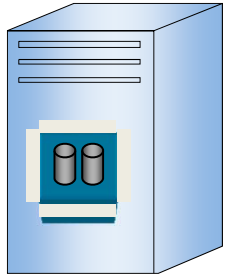
Evolving With Technology Changes



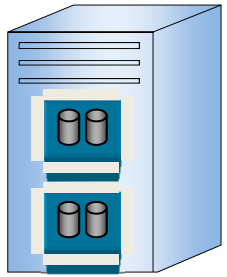
GPU



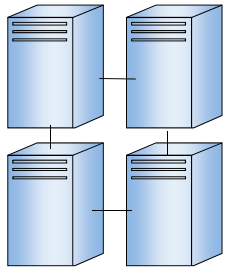
Single processor



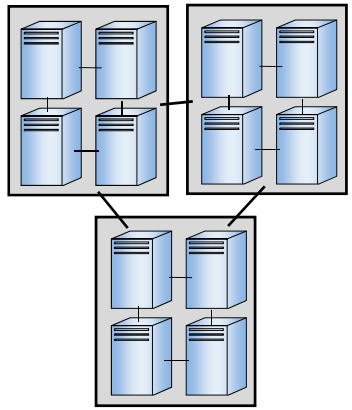
Multicore



Multiprocessor



Cluster



Grid,
Cloud

What's new in R2010b?

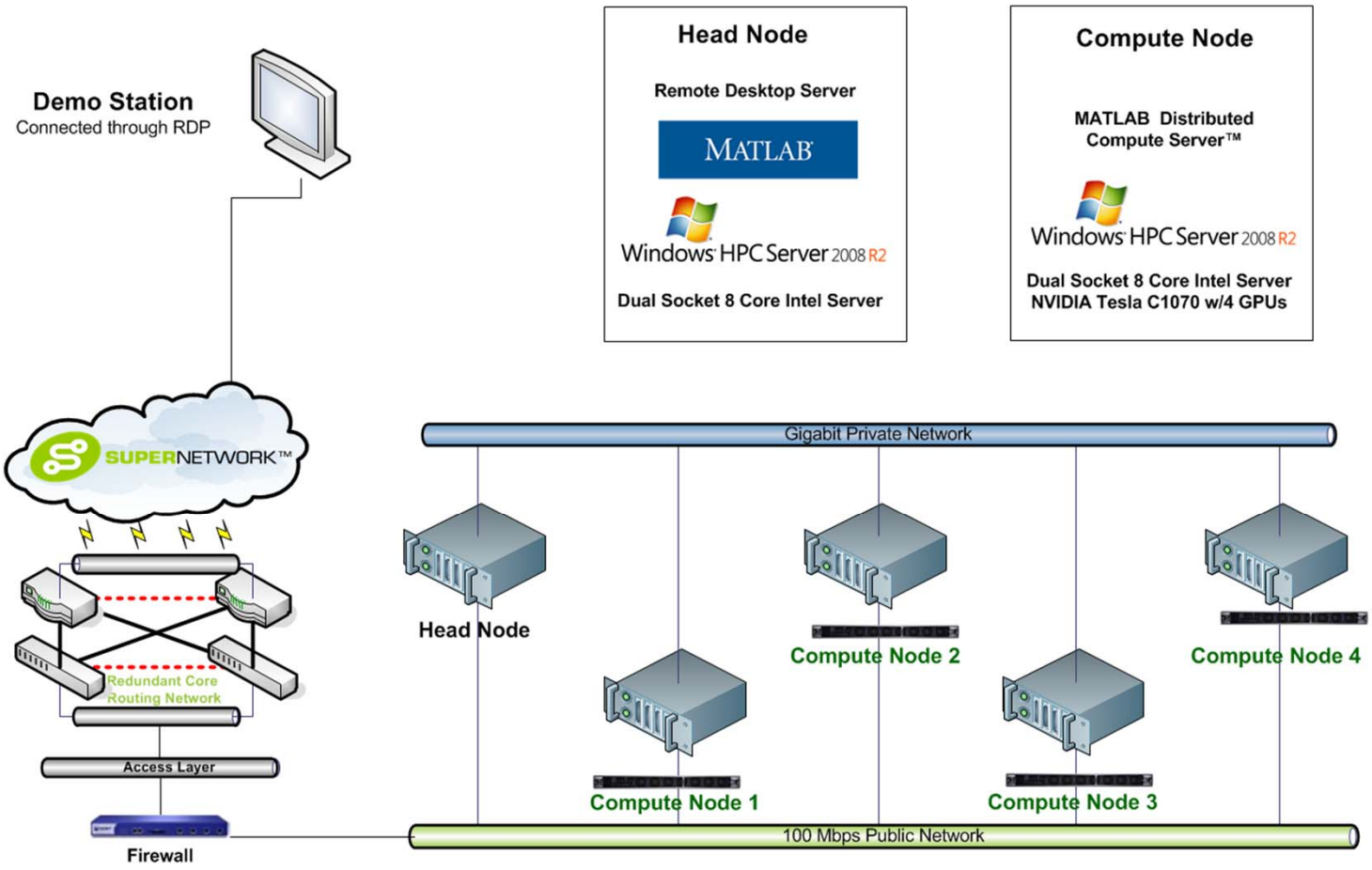
- Parallel Computing Toolbox
  - GPU support
  - Broader algorithm support (QR, rectangular \)
- MATLAB Distributed Computing Server
  - GPU support
  - Run as user with MathWorks job manager
  - Non-shared file system support
- Simulink®
  - Real-Time Workshop® support with PCT and MDCS

GPU Functionality

- Call GPU(s) from MATLAB or toolbox/server worker
- Support for CUDA 1.3 enabled devices
- GPU array data type
  - Store arrays in GPU device memory
  - Algorithm support for over 100 functions
  - Integer and double support
- GPU functions
  - Invoke element-wise MATLAB functions on the GPU
- CUDA kernel interface
  - Invoke CUDA kernels directly from MATLAB
  - No MEX programming necessary

Demo hardware

MathWorks Microsoft peer1 hosting



Example:

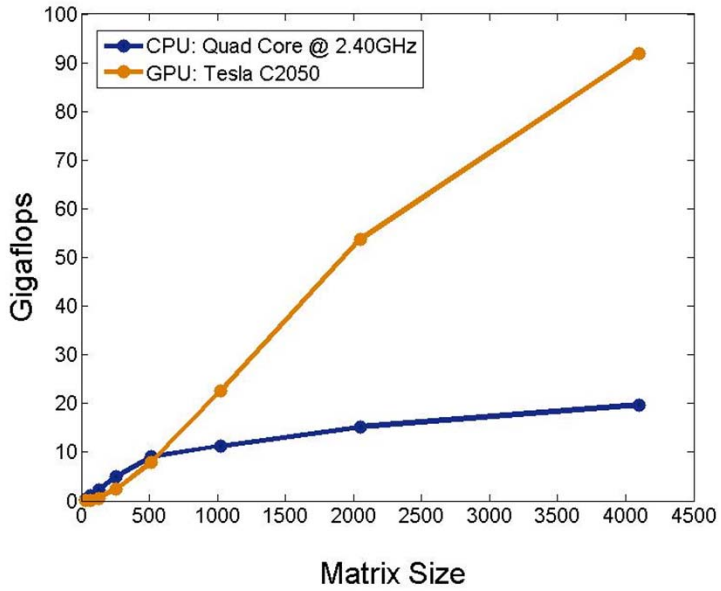
GPU Arrays

```
>> A = someArray(1000, 1000);
>> G = gpuArray(A); % Push to GPU memory
…
>> F = fft(G);
>> x = G\b;
…
>> z = gather(x); % Bring back into MATLAB
```

GPUArray Function Support

- >100 functions supported
  - `fft, fft2, ifft, ifft2`
  - Matrix multiplication (`A*B`)
  - Matrix left division (`A\b`)
  - LU factorization
  - `'  .'`
  - `abs, acos, …, minus, …, plus, …, sin, …`
- Key functions not supported
  - `conv, conv2, filter`
  - indexing

GPU Array benchmarks



A\b*	Tesla C1060	Tesla C2050 (Fermi)	Quad-core Intel CPU	Ratio (Fermi:CPU)
Single	191	250	48	5:1
Double	63.1	128	25	5:1
Ratio	3:1	2:1	2:1	

* Results in Gflops, matrix size 8192x8192. Limited by card memory.
Computational capabilities not saturated.

GPU Array benchmarks

MTIMES	Tesla C1060	Tesla C2050 (Fermi)	Quad-core Intel CPU	Ratio (Fermi:CPU)
Single	365	409	59	7:1
Double	75	175	29	6:1
Ratio	4.8:1	2.3:1	2:1	

FFT	Tesla C1060	Tesla C2050 (Fermi)	Quad-core Intel CPU	Ratio (Fermi:CPU)
Single	50	99	2.29	43:1
Double	22.5	44	1.47	30:1
Ratio	2.2:1	2.2:1	1.5:1	

Example:

arrayfun: Element-Wise Operations

```
>> y = arrayfun(@foo, x); % Execute on GPU
```

```
function y = foo(x)
y =  1 + x.*(1 + x.*(1 + x.*(1 + ...
 x.*(1 + x.*(1 + x.*(1 + x.*(1 + ...
 x.*(1 + x./9)./8)./7)./6)./5)./4)./3)./2);
```

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/336025232015010112