Multi-GPU Support for MATLAB® Using Jacket

www.accelereyes.com

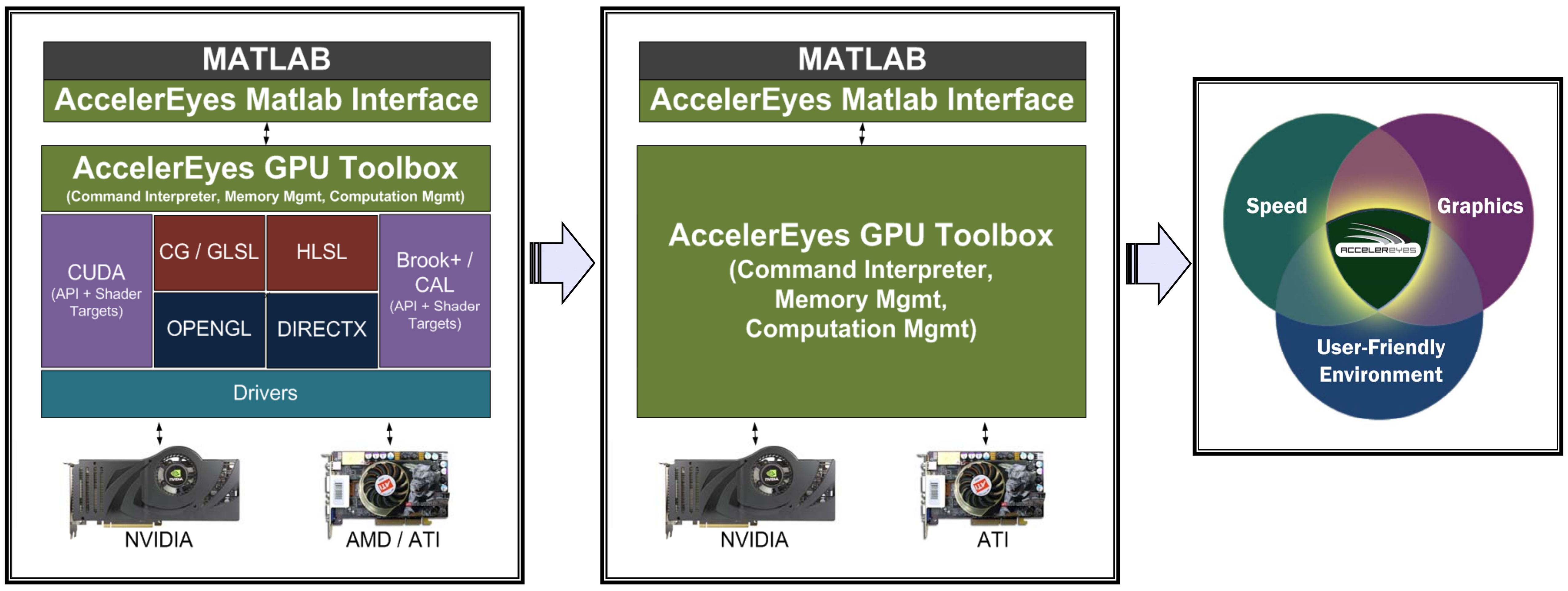
Gallagher Pryor, James Malcolm, John Melonakos, Tauseef ur Rehman
{gallagher.pryor@, malcolm@, john.melonakos@, tauseef.rehman@}accelereyes.com
75 5th Street, Suite 204, Atlanta, GA, USA

Abstract

In this poster, we present new features of Jacket: The GPU Engine for MATLAB® that enable scientific computing on GPUs. These include enhanced support for data and task parallel algorithms using 'gfor' and multi-GPU programming constructs that work seamlessly with the MATLAB's Parallel Computing Toolbox (PCT) and Distributed Computing Server (DCS), as well as sparse support for BLAS functions including point-wise matrix arithmetic and matrix-vector multiplications.

Jacket is a software platform for the rapid development of general purpose GPU (GPGPU) computing applications within the MATLAB computing environment. In addition to providing low-level GPGPU graphics and computing capabilities within MATLAB, Jacket includes a JIT compilation mechanism within the MATLAB environment that translates MATLAB code to GPGPU assembly code on demand..

Jacket Architecture

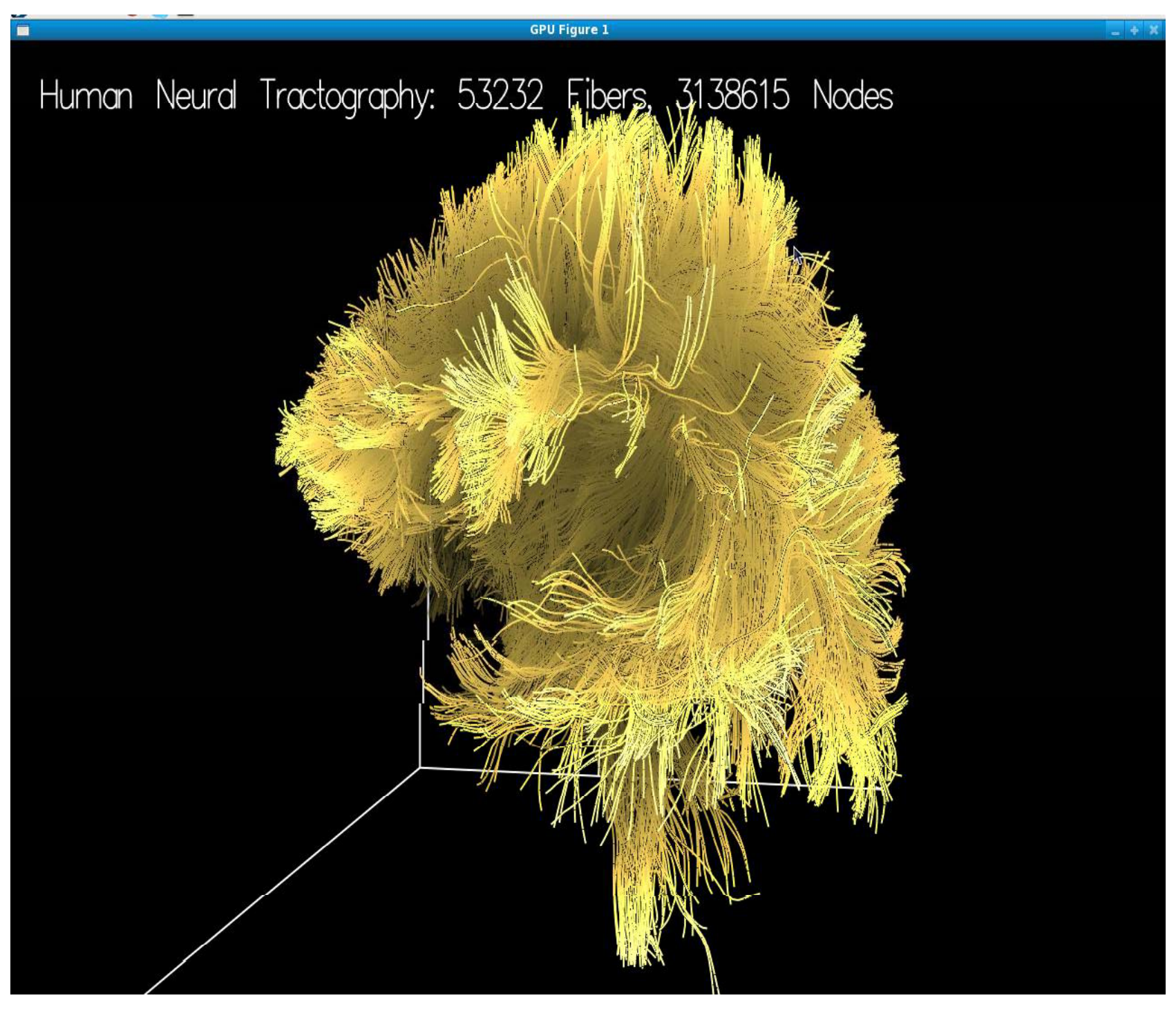


Benefit to Users

- ➢ ***Speed***: Jacket-enabled programs tap into the parallel computing power of the GPU. The GPU version of common programs achieves great speed improvements over equivalent CPU versions.
- ➢ ***User Friendly***: Jacket is not another GPU API. Rather it GPU-enables standard MATLAB code. Jacket users benefit from the user-friendly MATLAB environment.
- ➢ ***Visualization***: Jacket's **Graphics Toolbox** brings the full power of OpenGL graphics to MATLAB code.

Benefit to Projects

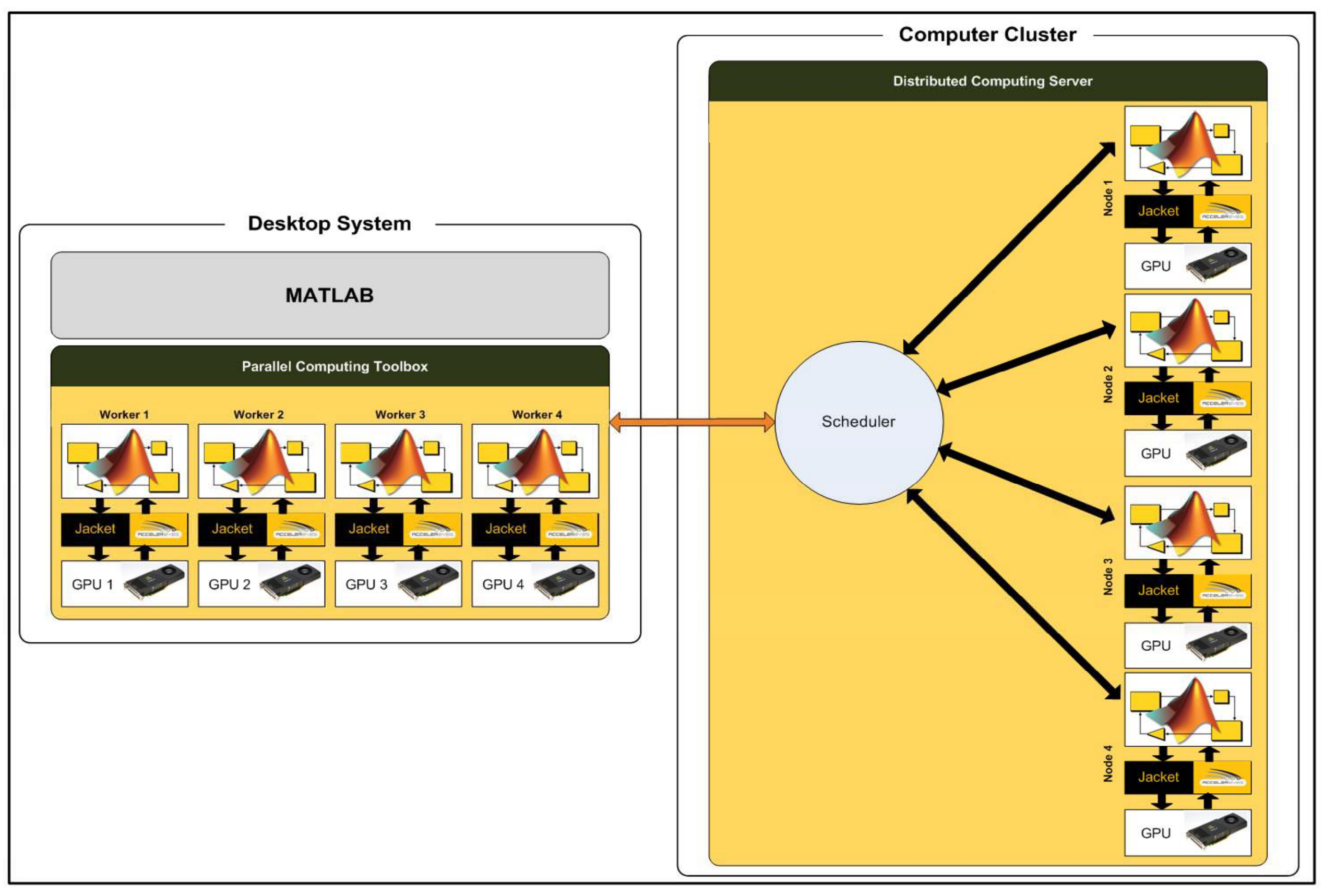
- ➢ ***Rapid Prototyping***: Now you can leverage the rapid prototyping benefits of MATLAB in your GPU programming.
- ➢ ***GPU Taste Testing***: AccelerEyes technology enables you to taste test the benefits of the GPU before investing a lot of time and money in expensive developmental programs.
- ➢ ***Eliminating the Port***: Jacket eliminates the port. Your MATLAB code can runs in real time by leveraging the power of GPUs within the MATLAB environment.
- ➢ ***Embedded GPUs***: With AccelerEyes' upcoming Runtime Build Target for GPUs, you will be able to program embedded GPUs for your application.

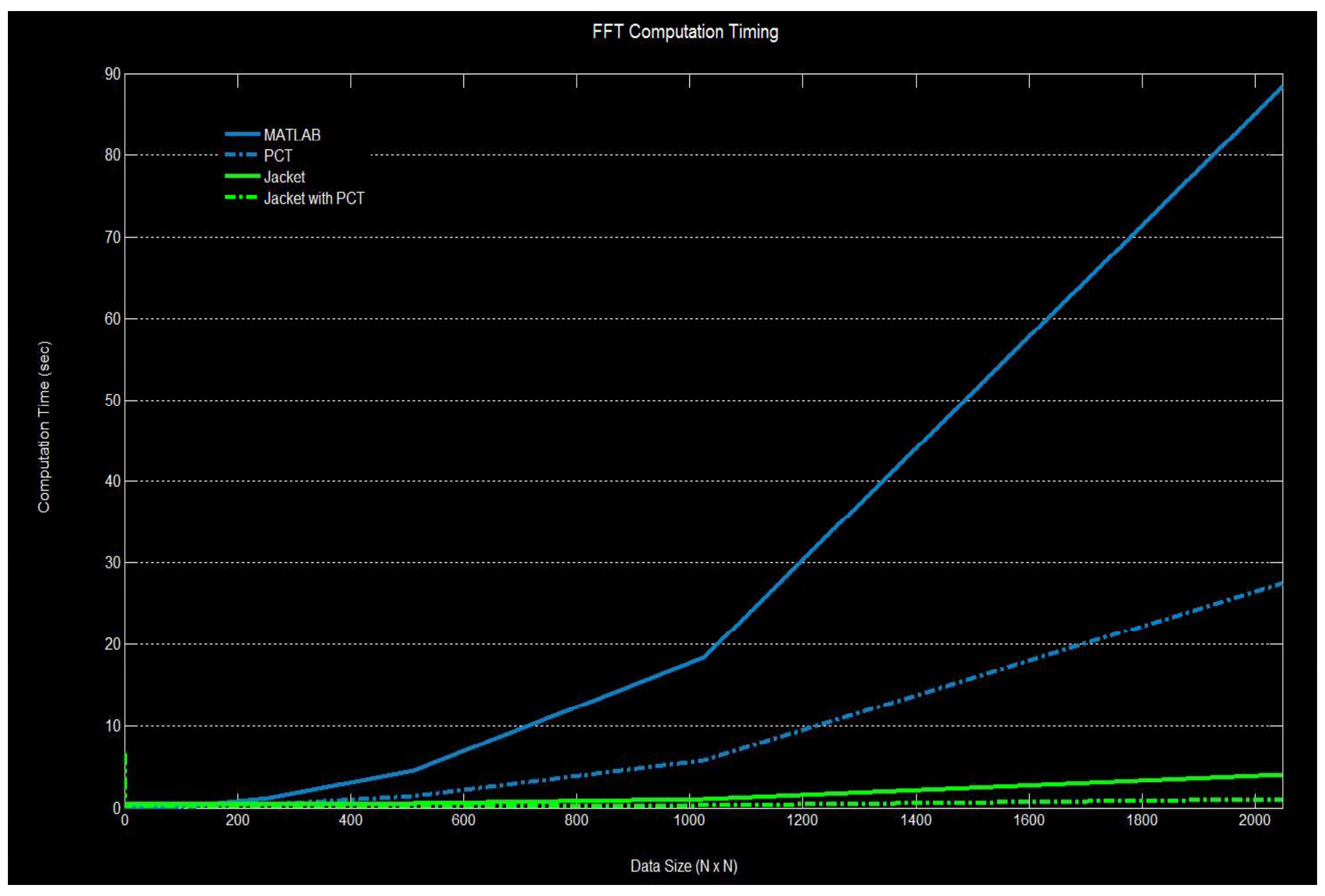


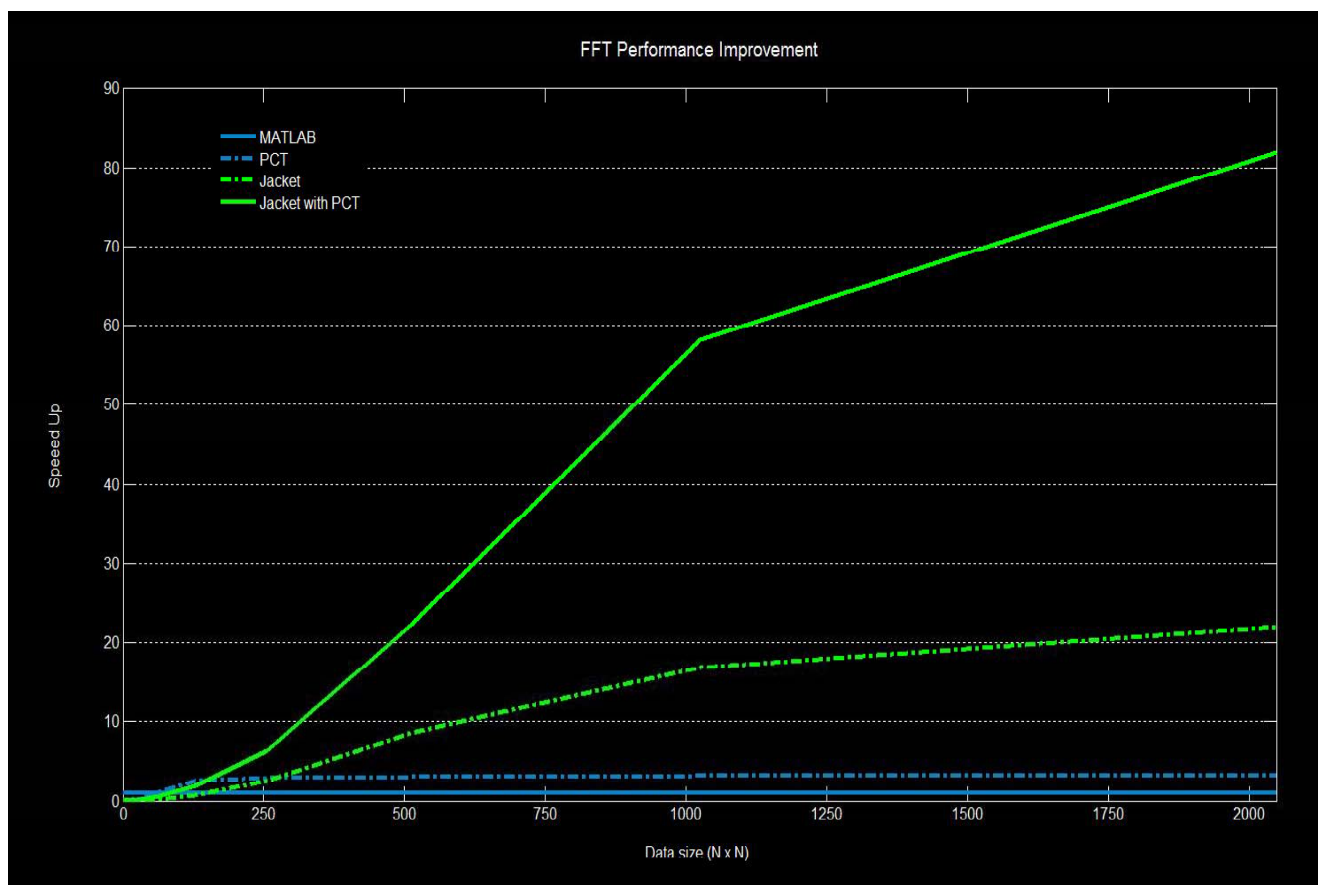
Multi-GPU Support

Jacket 1.2 now includes functionality to seamlessly utilize multiple GPUs either in the same machine or across a network via the MATLAB Parallel Computing Toolbox and the MATLAB distributed Computing Server. With the simple addition of well known parallel constructs such as **PARFOR**, **SPMD**, or **co-distributed arrays**, pre-existing code may be dispatched across all GPUs and CPUs in a Personal Super Computer or a cluster. In many cases, little to no code revision is required to take advantage of this new parallel computing capability.

Jacket's new ability to span computation across multiple GPUs either locally or over a network allows for an unprecedented ability to transparently scale GPU and CPU computing resources. Additional GPUs added to a host may now be instantly utilized without a code modification - simply increment the number of MATLAB workers via the MATLAB command prompt. When a host is not capable of driving more GPUs, simply add a GPU to another host on the network and drive it via the MATLAB Distributed Computing Server. With the addition of Jacket, pre-existing CPU clusters may be upgraded through the installation of GPUs, significantly increasing the cluster's computational capability without investing in new development for specialized GPU code. MATLAB paired with Jacket 1.2 is the easiest and most scalable solution for GPU computing available.







Sparse Support

Sparse matrix multiplication is an important enhancement for Jacket, being introduced with version 1.2. Sparse matrices arise in many applications naturally, and effectively representing and manipulating such sparse structures is an important necessity. The following is now supported:

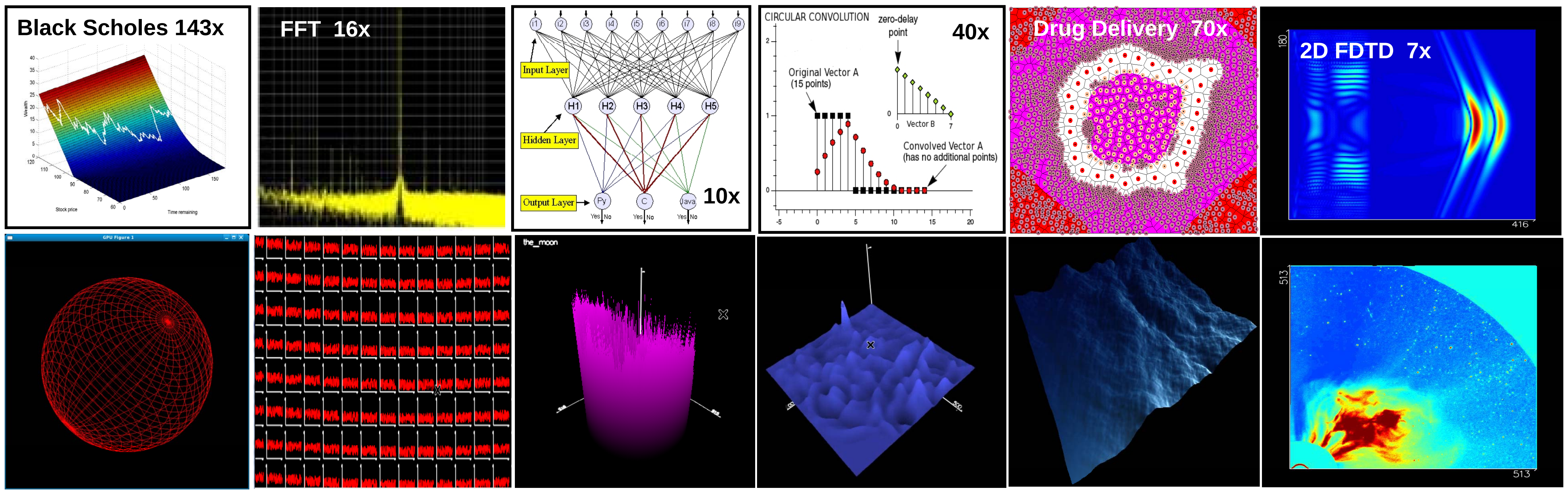
1. Creation of Sparse Matrices

```
A = gsingle( sprand( 10,10,0.2 ) );
```

2. Sparse Maxtrix x Dense Matrix/Vector

```
X = grand( 10 );
B = A × X;
```

Jacket Applications


Black Scholes 143x
Wealth
Stock price
Time remaining
FFT 16x
Input Layer
Hidden Layer
Output Layer
H1
H2
H3
H4
H5
Py
C
Java
Yes
No
10x
CIRCULAR CONVOLUTION
zero-delay point
40x
Original Vector A (15 points)
Vector B
Convolved Vector A (has no additional points)
Drug Delivery 70x
2D FDTD 7x
GPU Figure 1
the_moon