What is GPU Computing?

Computing with CPU + GPU

Heterogeneous Computing
Computation Discontinuity

- NVIDIA GPU
- Intel CPU

Double Precision debut

Gflops (log scale)

Tesla 8-series

Tesla 10-series

Intel Xeon Quad-core 3 GHz

Intel Core2 Dual-core 3.0 GHz

Intel Pentium 4 Dual-core 3.0 GHz

Intel Pentium 4 3.2 GHz

Double Precision debut
CUDA Parallel Computing Architecture

- Parallel computing architecture and programming model
- Massively parallel, many core architecture
- Architected to natively support all computational interfaces (standard languages and APIs)
NVIDIA Tesla 10-Series GPU

Massively parallel, many core architecture

240 Processor Cores

1 Teraflops - 1,000 times Cray X-MP

IEEE Compliant Double Precision Floating Point

Designed for Scientific Computing
CUDA Facts

- 900+ Research Papers
- 125+ universities teaching CUDA

www.NVIDIA.com/CUDA

- 200+ papers and applications
- 110 Million CUDA-Enabled GPUs
- 60,000+ Active Developers
• Who is AccelerEyes?
  – AccelerEyes is a MathWorks partner
  – Simple software for visual computing
Background

• Who is AccelerEyes?
  – AccelerEyes is a MathWorks partner
  – Simple software for visual computing

• What is Jacket?
  – GPU engine for MATLAB
  – CUDA powered language extension
Background

• Who is AccelerEyes?
  – AccelerEyes is a MathWorks partner
  – Simple software for visual computing

• What is Jacket?
  – GPU engine for MATLAB
  – CUDA powered language extension

• Why Jacket?
  – Challenges in technical computing
  – Low-cost speed, high-value graphics
  – Increased productivity
MATLAB Options

- CPU Solutions (blue arrows)
  - MATLAB and the Parallel Computing toolbox enable PC and clustered MATLAB computing

- GPU Solutions (green arrows)
  - Jacket enables CUDA MATLAB computing
Jacket combines the speed of CUDA and the graphics of the GPU with the user friendliness of MATLAB.
Functionality

- Generators: geye, gones, gzeros
- Element-wise: +, *, -, /
- Reductions: sum, min, max ...
- Indexing: subscripted referencing / subscripted assignment
- Linear Algebra: matrix multiply, ...
- FFT: fft, ifft, fftn, ifftn
- Filtering: filter, filter2, convn
- Interpolation: interp2
- Parallel for-loops: gfor
Kernel Benchmarks

**SGEMM Kernel**
- **Matrix Size**
  - 2048x2048: 48 secs (Tesla C1060), 2637 secs (Intel Core 2 Duo (2.0 GHz)), 54x Speedup
  - 1024x1024: 1.1 secs (Tesla C1060), 117 secs (Intel Core 2 Duo (2.0 GHz))
  - 512x512: 1.3 secs (Tesla C1060), 12.7 secs (Intel Core 2 Duo (2.0 GHz))

**FFT Kernel**
- **Vector Length**
  - 1024x1024: 1.1 secs (Tesla C1060), 17.7 secs (Intel Core 2 Duo (2.0 GHz)), 16x Speedup
  - 512x512: 0.43 secs (Tesla C1060), 3.6 secs (Intel Core 2 Duo (2.0 GHz))
  - 256x256: 0.28 secs (Tesla C1060), 0.91 secs (Intel Core 2 Duo (2.0 GHz))

*Simple Software. Powerful Visual Computing.*
Genomics Example

Center for Disease Control

Speedup – 3X on GPU

- Establishing a network of coordinated mutations in Hepatitis C virus genome
- 114 sequences of sub-genotype 1b in HCV
- 226 sequences after ancestral tree reconstruction
- 10,000 nucleotides
- 700,000 pairs of positions of interest
Medical Image Processing

Harvard, Stanford
- Image reconstruction, segmentation, registration

Howard Hughes Medical Institute
- Realtime monitoring of animal experiments

Emory University, Fairchild Imaging
- Cellular imaging, image stacks and 3d models
Image Thresholding

Speedup: 20X on 512x512

Drug Delivery Example

Georgia Tech – MIBLAB
Speedup – 70X


Spencer Technologies

Speedup – 12X on GPU

• Realtime system
• 300ms duty cycle for acquisition, data transfer, computation, & visualization
• Custom CUDA code
/* each element gets its index */
static __global__ void kernel(float *d_out, float *d_in)
{
  int x = blockIdx.x * blockDim.x + threadIdx.x;
  d_out[x] = d_in[x] + x;
}

void mexFunction(int nlhs, mxArray *plhs[], int nrhs, const mxArray *prhs[])
{
  // attach to jacket /
  CUSContext *ctx = (CUSContext *) (unsigned int)mexGetScalar(prhs[0]);
  cuCtxAttach(ctx, 0);

  /* get device pointer of gsingle */
  float *d_in = (float *) (unsigned int)mexGetScalar(prhs[1]);
  /* run kernel to initialize 10 elements */
  float *d_out;
  cudaMalloc(void **d_out, 10*sizeof(float));
  kernel<<<1,16>>>(d_out, d_in);
  /* pull back to cpu and print */
  float h_out[10];
  cudaMemcpy(h_out, d_out, 10*sizeof(float), cudaMemcpyDeviceToHost);
  for (int i = 0; i < 10; i++)
    printf("%f", h_out[i]);

  // detach from jacket /
  cuCtxDetach(ctx);

  /* return device pointers */
  plhs[0] = mxCreateDoubleScalar((unsigned int)d_out);
}
Graphics Toolbox

Jacket includes the Graphics Toolbox

- True visual computing
- OpenGL API in MATLAB
- Interactive OpenGL
- Key functions: gsurf, gimage, gscatter3, gplot, ...
- Visualization scripts are open and modifiable.

Some Jacket Customers

- National Cancer Institute
- Siemens
- Harvard Medical School
- Northrop Grumman
- GE
- Motorola
- Microsoft Research
- Toyota
- Google
- NASA
- Philips
- Ames Research Center
- Carnegie Mellon
- L3 Communications
- Raytheon
- Johns Hopkins University
- Deutsche Bank
- TUG
- eSolar
- Sentilla
- Stanford University
Roadmap for New Features

- more gfor
- gdouble
- multi-GPU support (for clusters of GPUs)
- LAPACK (eig, inv, etc.)
- signal processing
- image processing (and computer vision)
- Simulink® on the GPU
- statistical functions
- handle graphics
- lots of other MATLAB functions (finance, biology, etc.)
Tesla GPU Computing Products

Built for High Performance Computing
## Tesla GPU Computing Products

<table>
<thead>
<tr>
<th></th>
<th>Tesla S1070 1U System</th>
<th>Tesla C1060 Computing Board</th>
<th>Tesla Personal Supercomputer (4 Tesla C1060s)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GPUs</strong></td>
<td>4 Tesla GPUs</td>
<td>1 Tesla GPU</td>
<td>4 Tesla GPUs</td>
</tr>
<tr>
<td><strong>Single Precision Perf</strong></td>
<td>4.14 Teraflops</td>
<td>933 Gigaflops</td>
<td>3.7 Teraflops</td>
</tr>
<tr>
<td><strong>Double Precision Perf</strong></td>
<td>346 Gigaflops</td>
<td>78 Gigaflops</td>
<td>312 Gigaflops</td>
</tr>
<tr>
<td><strong>Memory</strong></td>
<td>4 GB / GPU</td>
<td>4 GB</td>
<td>4 GB / GPU</td>
</tr>
</tbody>
</table>

![Tesla GPU Computing Products](image)
**Tesla Personal Supercomputer: Cluster Perf**

**Supercomputing Performance**
- 960 cores. 4 TeraFlops
- Performance of a 64-node CPU cluster

**Personal**
- One researcher, one supercomputer
- Plugs into standard power strip

**Accessible**
- Program in C for Windows, Linux
Tesla S1070: Supercharge your cluster

- Hess
- Chevron
- Petrobras
- NCSA
- CEA
- Tokyo Tech
- JFCOM
- SAIC
- Federal
- Motorola
- Kodak
- BNP Paribas
- University of Heidelberg
- University of Illinois
- University of North Carolina
- Max Planck Institute
- Rice University
- University of Maryland
- Eotvas University
- University of Wuppertal
- Chinese Academy of Sciences
- National Taiwan University
### $5 Million Cluster: Lower Power, Higher Perf

**CPU 1U Server**

<table>
<thead>
<tr>
<th>2 Quad-core Xeon CPUs: 8 cores</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.17 Teraflop (single)</td>
</tr>
<tr>
<td>0.08 Teraflop (double)</td>
</tr>
</tbody>
</table>

- 1819 CPU servers
- 310 Teraflops (single)
- 155 Teraflops (double)
- Total area 16K sq feet
- Total 1273 KW

**8 CPU Cores + 4 GPUs = 968 cores**

| 4.14 Teraflops (single)       |
| 0.346 Teraflop (double)       |

- 455 CPU servers
- 455 Tesla systems
- 1961 Teraflops (single)
- 196 Teraflops (double)
- Total area 9K sq feet
- Total 682 KW

- 50% fewer systems
- 6x more perf
- 40% smaller
- ½ the power

**Tesla 1U System**

- 40% smaller
- ½ the power
## 5000+ Customers / ISVs

<table>
<thead>
<tr>
<th>Life Sciences &amp; Medical Equipment</th>
<th>Productivity / Misc</th>
<th>Oil and Gas</th>
<th>EDA</th>
<th>Finance</th>
<th>CAE / Mathematical</th>
<th>Communication</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max Planck</td>
<td>GE Healthcare</td>
<td>Hess</td>
<td>Synopsys</td>
<td>Symcor</td>
<td>AccelerEyes</td>
<td>Nokia</td>
</tr>
<tr>
<td>FDA</td>
<td>Siemens</td>
<td>TOTAL</td>
<td>Nascentric</td>
<td>Level 3</td>
<td>MathWorks</td>
<td>RIM</td>
</tr>
<tr>
<td>Robarts Research</td>
<td>Techniscan</td>
<td>CGG/Veritas</td>
<td>Gauda</td>
<td>SciComp</td>
<td>Wolfram</td>
<td>Philips</td>
</tr>
<tr>
<td>Medtronic</td>
<td>Boston Scientific</td>
<td>Chevron</td>
<td>CST</td>
<td>Hanweck</td>
<td>National Instruments</td>
<td>Samsung</td>
</tr>
<tr>
<td>AGC</td>
<td>Eli Lilly</td>
<td>Headwave</td>
<td>Agilent</td>
<td>Quant</td>
<td>Ansys</td>
<td>LG</td>
</tr>
<tr>
<td>Evolved machines</td>
<td>Silicon Informatics</td>
<td>Acceleware</td>
<td></td>
<td>Catalyst</td>
<td>Access Analytics</td>
<td>Sony</td>
</tr>
<tr>
<td>Smith-Waterman</td>
<td>Stockholm Research</td>
<td>Seismic City</td>
<td></td>
<td>RogueWave</td>
<td>Tech-x</td>
<td>Ericsson</td>
</tr>
<tr>
<td>DNA sequencing</td>
<td>Harvard</td>
<td>P-Wave</td>
<td></td>
<td></td>
<td>RIKEN</td>
<td>NTT DoCoMo</td>
</tr>
<tr>
<td>AutoDock</td>
<td>Delaware</td>
<td>Seismic</td>
<td></td>
<td></td>
<td>SOFA</td>
<td>Mitsubishi</td>
</tr>
<tr>
<td>NAMD/VMD</td>
<td>Pittsburg</td>
<td>Imaging</td>
<td></td>
<td></td>
<td>Renault</td>
<td>Hitachi</td>
</tr>
<tr>
<td>Folding@Home</td>
<td>ETH Zurich</td>
<td>Mercury</td>
<td></td>
<td></td>
<td>Boeing</td>
<td>Radio</td>
</tr>
<tr>
<td>Howard Hughes</td>
<td>Institute Atomic Physics</td>
<td>Computer</td>
<td></td>
<td></td>
<td></td>
<td>Research Laboratory</td>
</tr>
<tr>
<td>Medical</td>
<td></td>
<td>ffA</td>
<td></td>
<td></td>
<td></td>
<td>US Air Force</td>
</tr>
<tr>
<td>CRIBI Genomics</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

CUDA
More Information

- **Tesla main page**
  - [http://www.nvidia.com/tesla](http://www.nvidia.com/tesla)

- **Vertical Solutions**

- **CUDA Zone**
  - CUDA Tutorials, Applications

- **Hear from Developers**
  - [http://www.youtube.com/nvidiatesla](http://www.youtube.com/nvidiatesla)

---

Download Jacket Now

- [http://www.accelereyes.com](http://www.accelereyes.com)

Further Jacket Questions

- [http://www.accelereyes.com/forums](http://www.accelereyes.com/forums)
- [http://www.accelereyes.com/blog](http://www.accelereyes.com/blog)

---

Sumit Gupta
sumitg@nvidia.com

John Melonakos
john.melonakos@accelereyes.com