

VSCSE summer school - short course

Introduction to CUDA

Lecture I

Introduction to Many-Core Processors

Joshua A. Anderson

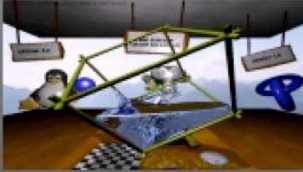
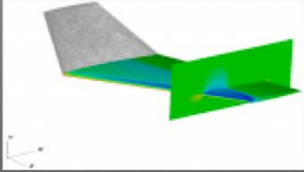
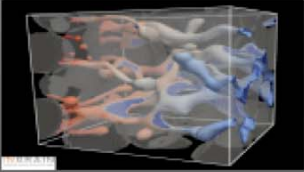
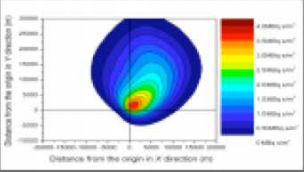
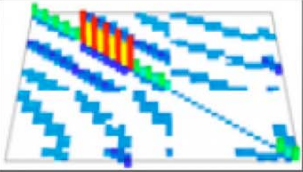
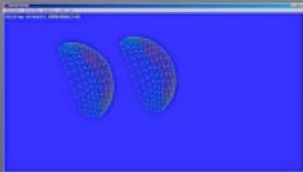
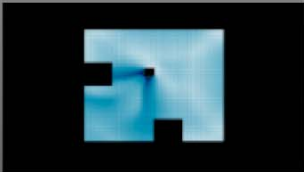
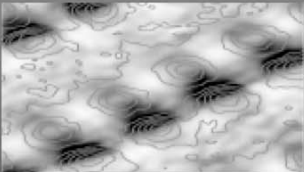
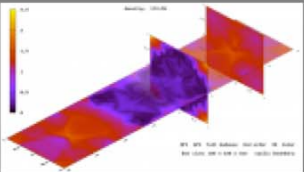
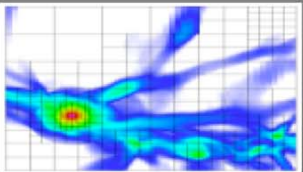

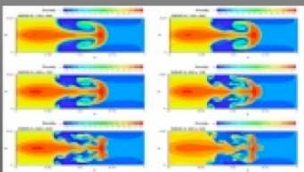

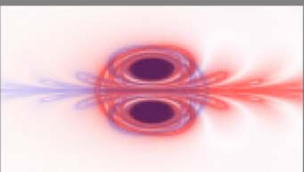

Modern video games demand copious processing power.

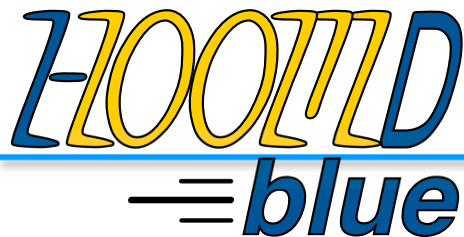


- **Millions** of pixels
- **Thousands** of calculations per pixel
- One **hundred** frames per second
- 100's of GFLOPS are needed

Applications using CUDA

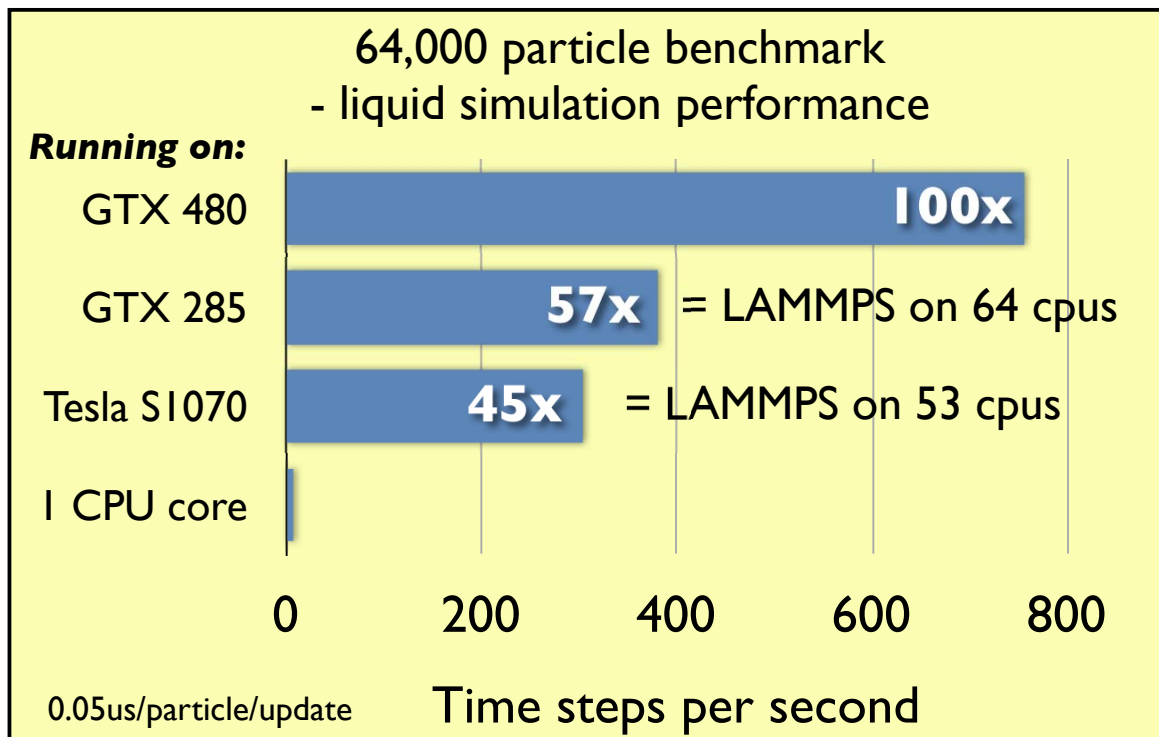
http://www.nvidia.com/object/cuda_apps_flash_new.html#

 <p>Realtime free surface fluid simulation and visuali...</p> <p>10 x</p>	 <p>ANDSolver</p> <p>100 x</p>	 <p>Multiphase flow in porous media</p> <p>100 x</p>	 <p>Stochastic Lagrangian Particle Model for Air Pollu...</p> <p>120 x</p>	 <p>Iterative CUDA</p> <p>10 x</p>
 <p>QView</p> <p>8 x</p>	 <p>Lattice-Boltzmann Simulation of the Shallow-Water ...</p> <p>8 x</p>	 <p>Accelerating Geo-Science and Engineering System Si...</p> <p>30 x</p>	 <p>nHD</p> <p>173 x</p>	 <p>GAMER: a GPU-Accelerated Adaptive Mesh Refinement ...</p> <p>12 x</p>
 <p>Incompressible Flow Computations on the NCSA Linco...</p> <p>50 x</p>	 <p>Acceleration of a Finite-Difference WENO Scheme fo...</p> <p>50 x</p>	 <p>Towards a multi-GPU solver for the three-dimension...</p> <p>16 x</p>	 <p>Optimization of FTLE Calculation</p> <p>1000 x</p>	 <p>FOLKI GPU</p> <p>100 x</p>



Open source:
<http://codeblue.umich.edu/hoomd-blue>

General purpose Many-particle Dynamics
fully implemented on GPU hardware



32 dual quad core CPU nodes w/ fast Infiniband

\$140,000



One desktop w/
4 GPUs



\$5000



Desktop GPU: Cost per performance



	GTX 285	Tesla C1060	GTX 480	Tesla C20X0
Processor elements	240	240	480	448
Peak compute (single)	1.062 TFLOPS	933 GFLOPS	1.3 TFLOPS	1.03 TFLOPS
Peak compute (double)	88.5 GFLOPs	77 GFLOPS	158 GFLOPS	515 GFLOPS
Memory Bandwidth	159 GB/s	102 GB/s	177 GB/s	144 GB/s
Memory size	1 GB	4 GB	1.5 GB	3 - 6 GB
Cost	\$330	\$1300	\$500	\$2500

2008-2009

2010

Workstation and datacenter GPUs

Build your own: http://www.nvidia.com/object/tesla_build_your_own.html
Pre-configured systems: http://www.nvidia.com/object/tesla_supercomputer_wtb.html

4-GPU Tesla Personal Supercomputers



AMAX ServMax PSC-2
Delivering cluster level computing performance to your desk – 250 times faster than traditional servers and workstations.

BUY NOW



Colfax CXT3000
Put the Power of Supercomputing in Your Hands
Own the computing power you need to visualize large data-sets, speed calculations, and solve problems impossible with current computing approaches.

BUY NOW

3-GPU Tesla Personal Supercomputers



AMAX's ServMax PSC
AMAX's ServMax PSC is a cluster in a box. It is optimized for scientific computing, delivering up to 15x cost savings and 15x lower power than traditional 1U rack-optimized servers.

BUY NOW



Colfax CXT3000i PSC
Configurable with up to 3 NVIDIA® Tesla™ C1060's and 1 NVIDIA® Quadro® FX5800 to deliver multi-Teraflops of peak performance, the CXT3000i delivers cluster level computing performance – right at your desktop.

BUY NOW



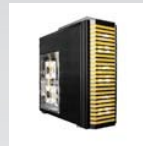
Hyperform HPCg A2401
The HPCg A2401 high-performance workstation by Silicon Mechanics features up to 3 NVIDIA Tesla C1060 GPUs, and outperforms a small traditional cluster.

BUY NOW



Microway's WhisperStation - PSC
incorporates multiple Tesla C1060 GPUs with Two CPUs (Quad Core Intel Xeon or AMD Opteron) and up to 32 GB DRAM. Storage options include individual disks or RAID storage.

BUY NOW



Penguin Computing Niveus HTX
Penguin Computing Niveus HTX provides 3TFLOPS of NVIDIA Tesla computing power in attractive desksize form factor.

BUY NOW



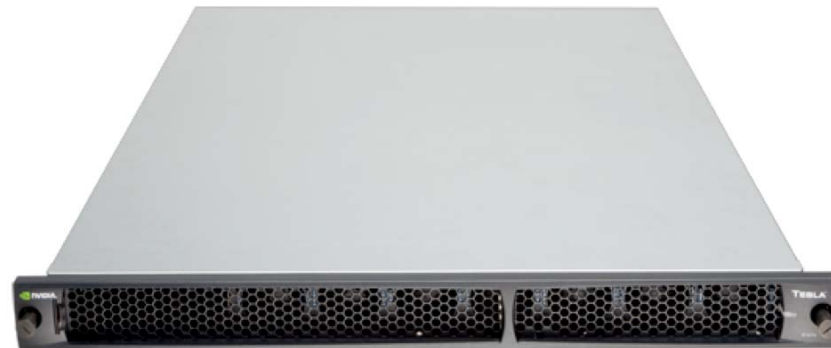
Velocity Micro ProMagix™ VSC Visual Supercomputing
Highly optimized, extreme performance, GPU accelerated visualization workstations for scientific computing.

BUY NOW

AC <http://iacat.uiuc.edu/resources/cluster/>



Tesla S1070



= 4x Tesla C1060 + power supply + fans

Large GPU clusters

- Lincoln - at NCSA
 - 1536 CPU cores
 - 384 Tesla GPUs
 - A TeraGrid resource

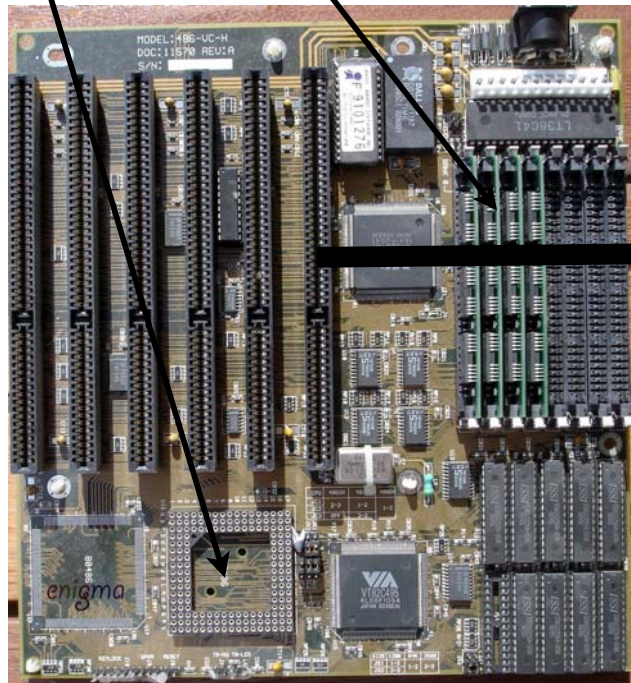
<http://www.ncsa.illinois.edu/UserInfo/Resources/Hardware/Intel64TeslaCluster/>

- TSUBAME - at Tokyo Tech
 - First Tesla equipped cluster on the Top500
- Many more
 - http://www.nvidia.com/object/cuda_clusters.html

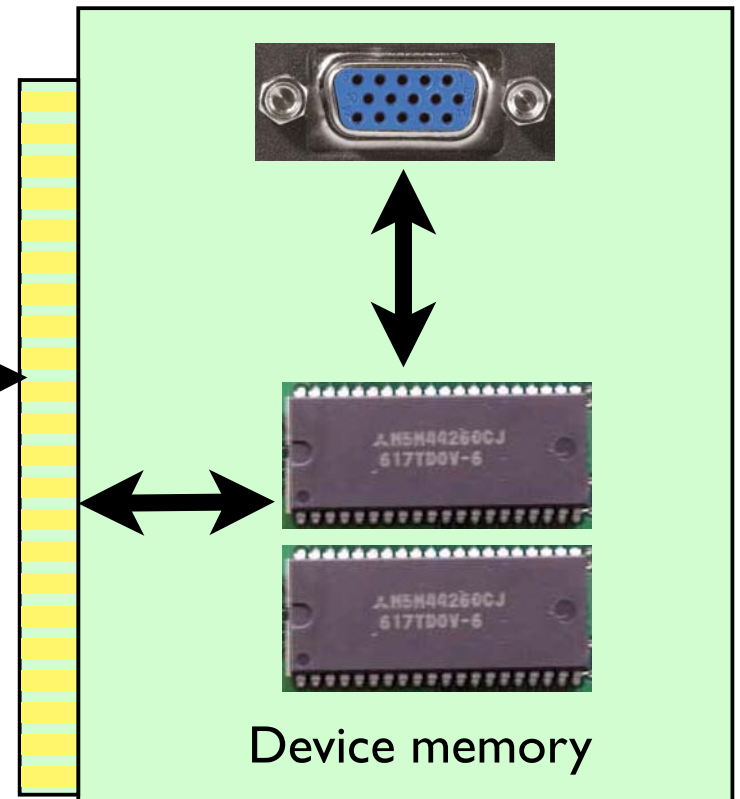


GPU prehistory (1981 - 1997)

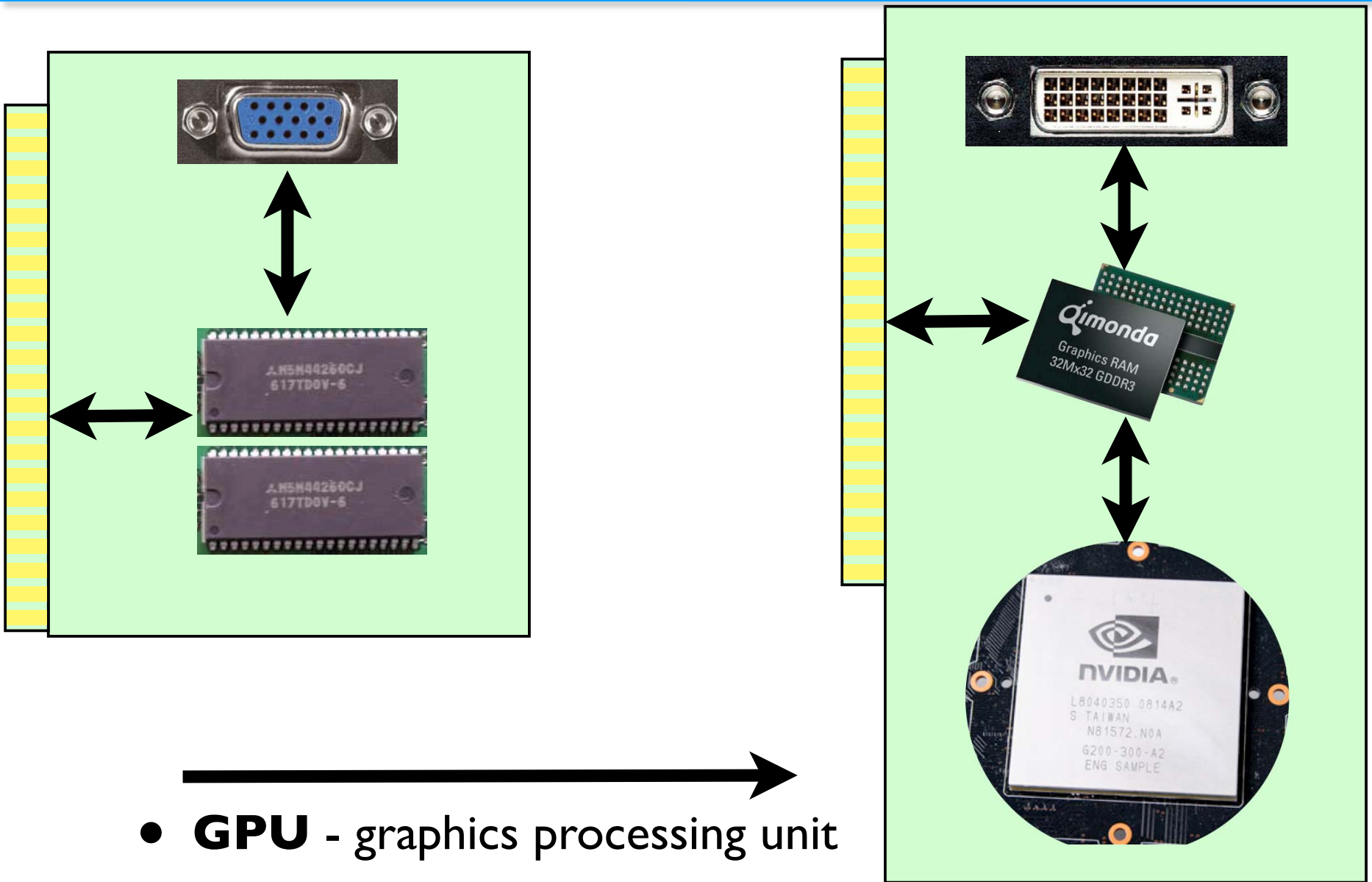
- **CPU** - central processing unit
- **Memory** - Semi-permanent storage for data



- **PCIe** - Point to point communications link. Used to connect the host to the device.



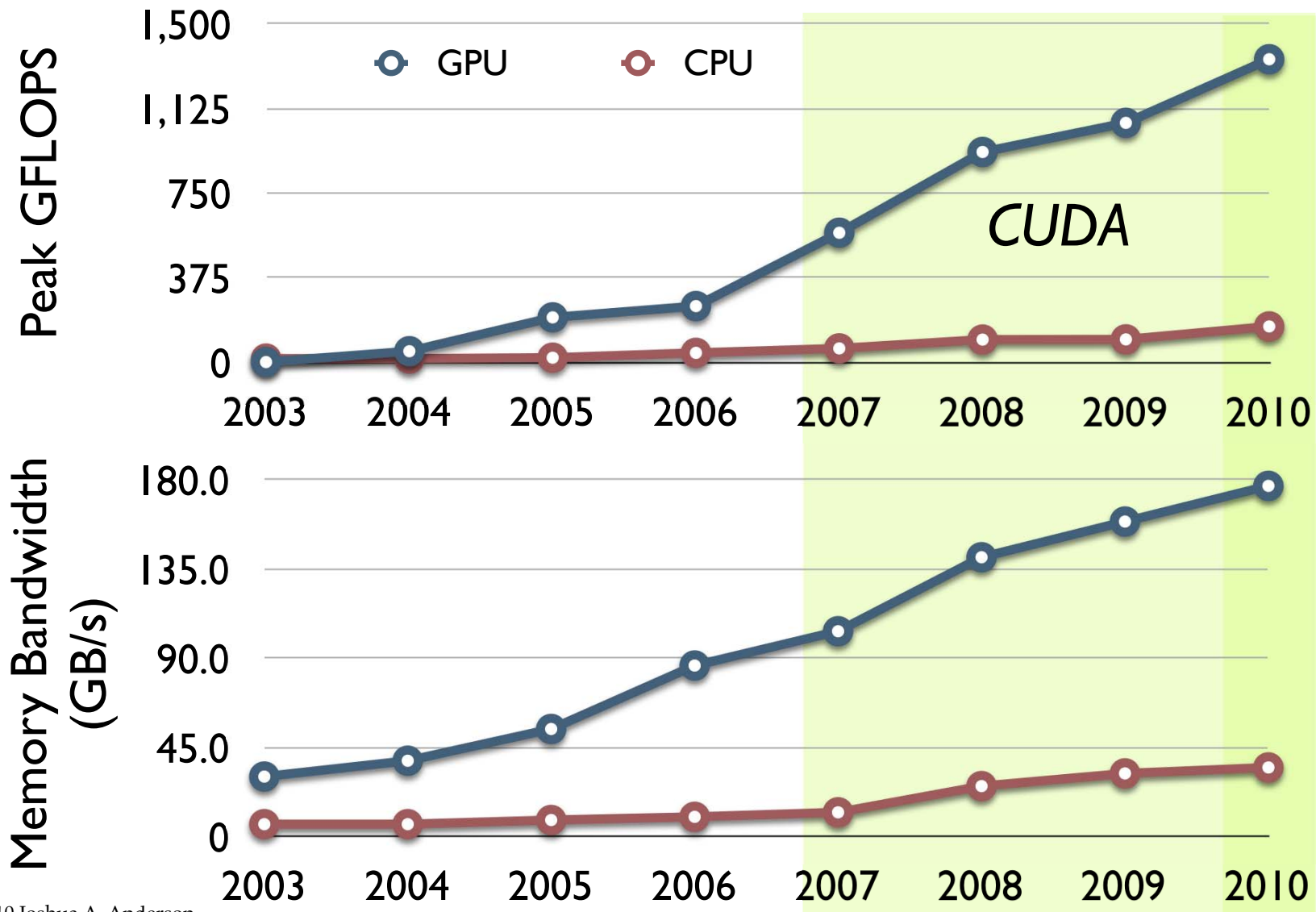
1997: 3DFX - the GPU is born



GPU performance in recent history

Performance of NVIDIA GPUs over time

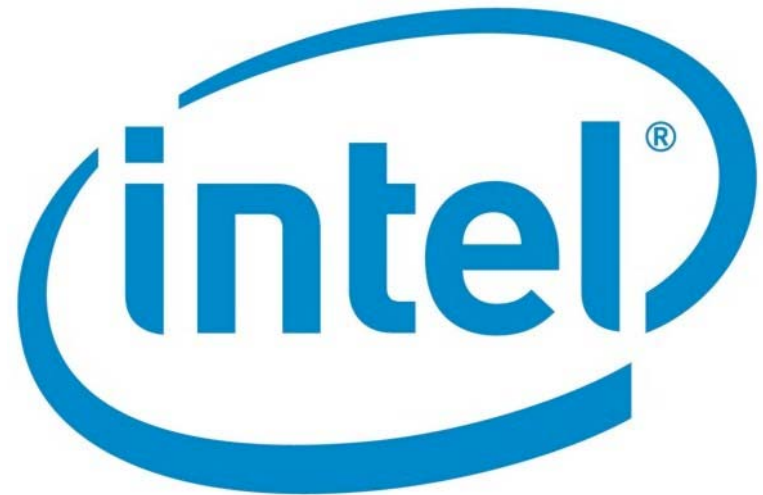
Fermi



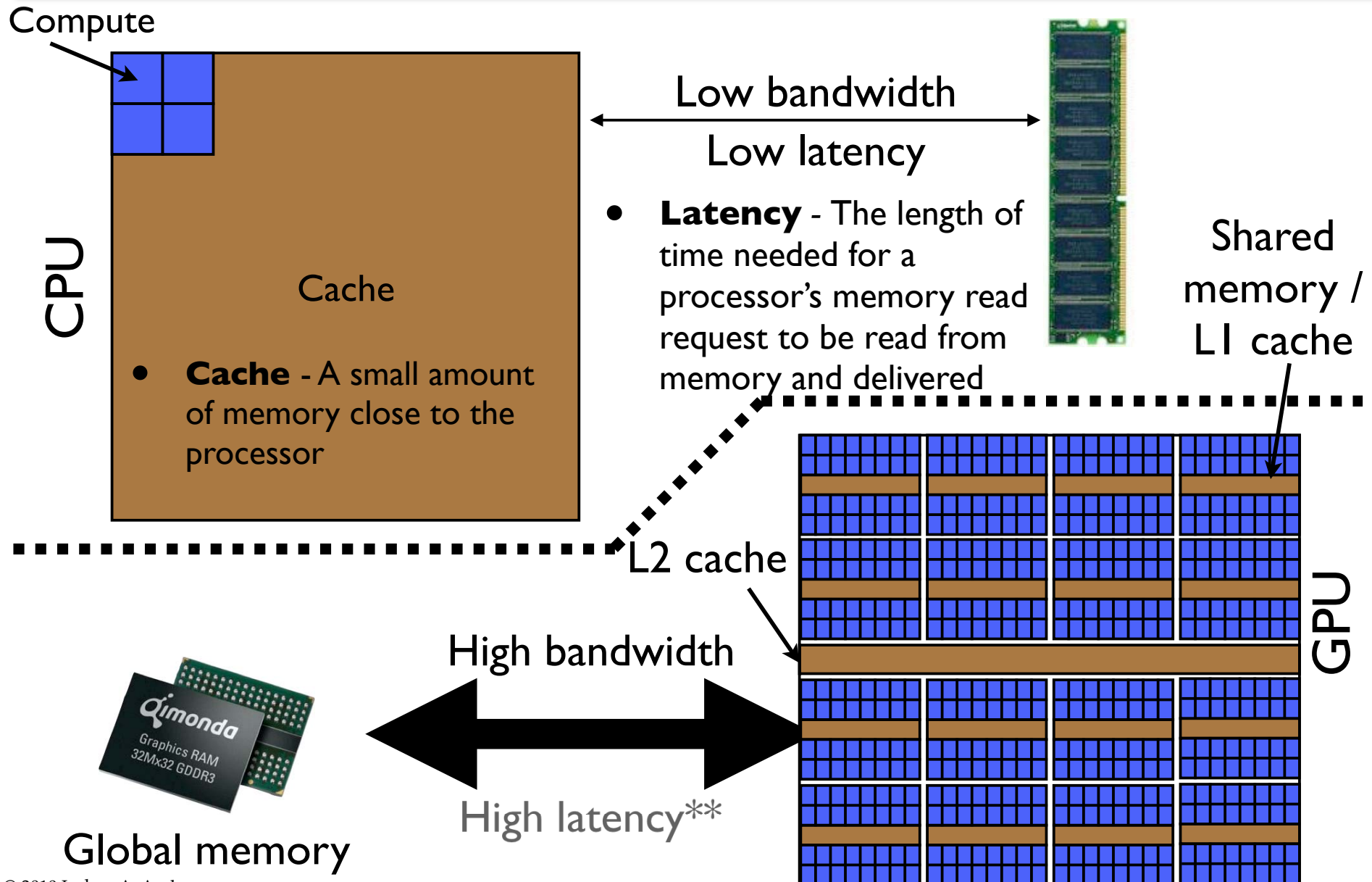
The industry competitors today



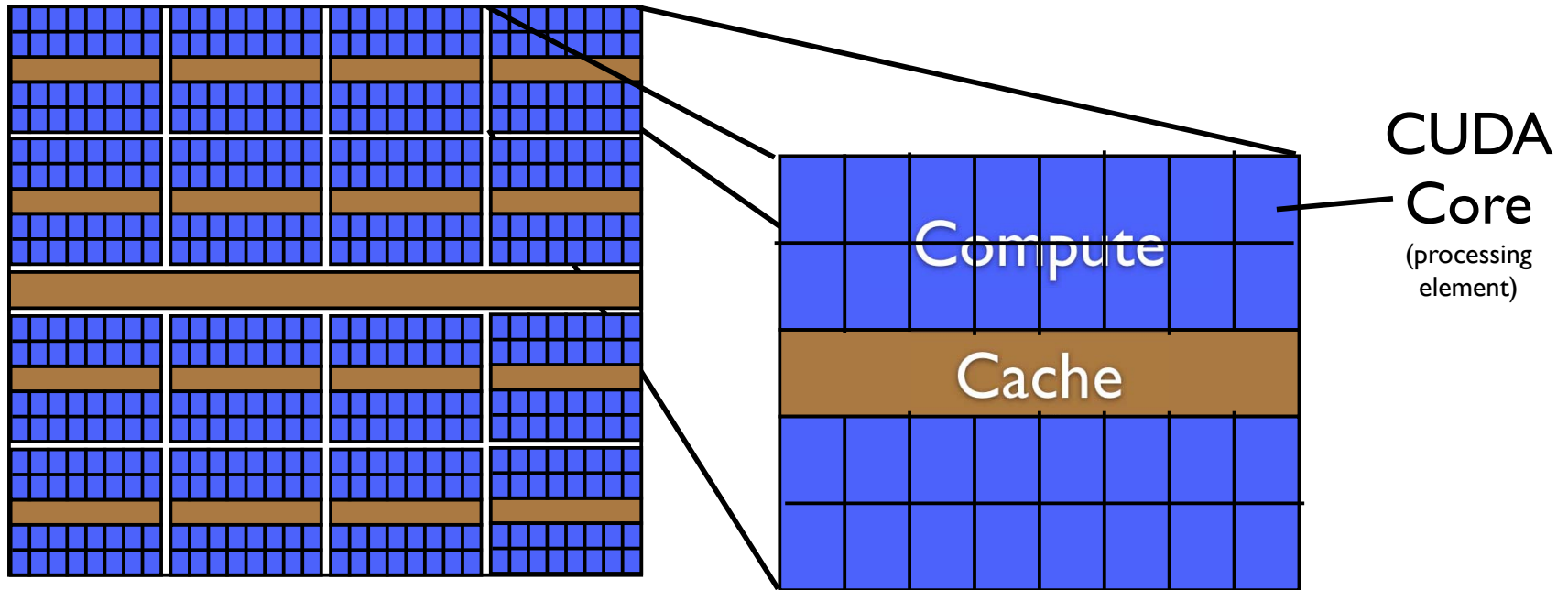
- **74.9 million** GPUs sold in Q1 2009
- Three big players
- Healthy competition and large demand => GPUs are **inexpensive** and **widely supported**



GPU architecture comparison



Multiprocessor capabilities



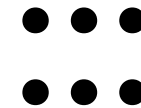
Each multiprocessor has:

CUDA cores	32
Total cache	64 kb
Registers	32,768
Floating point	IEEE-754 double
Number of active threads	1536

- **thread** - An independent sequence of operations (i.e. add, multiply, load, store, compare, branch...)
- **Register** - Temporary storage for the inputs and outputs of compute instructions.

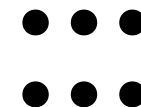
Wait, did you say *1536 active threads!*?

Example instruction stream



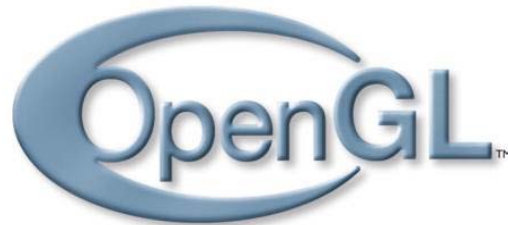
- Instructions are processed in **warps** of 32 threads
- any pattern of branches is handled by the **hardware**
- A multiprocessor is a huge **latency hiding** engine
- 16,384 registers => **no overhead** swapping contexts
- **SIMT**, not SIMD

warp 1	<i>memory load - sleeps</i>
warp 4	multiply
warp 2	add
warp 3	<i>memory load - sleeps</i>
warp 2	subtract
warp 1	<i>wakes up</i> - multiply
warp 4	subtract
warp 3	<i>wakes up</i> - subtract
warp 1	add



GPU programming environments

Graphics “only”



General purpose compute



OpenCL

Stream programming



Embedded
Metaprogramming
Language



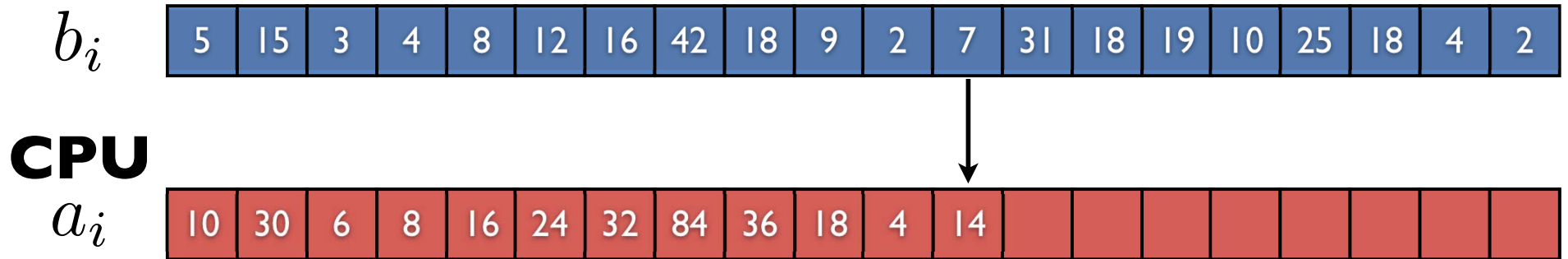
RAPIDMIND

- CUDA
 - Program in **C/C++**
 - **Fortran** support is available
 - **Very** easy to learn and use
 - Close enough to the hardware to obtain near peak performance
 - No limitations on scatter
 - Very good documentation and learning material
- OpenCL = CUDA-like programming model for a variety of different hardware platforms

Data-parallel execution model

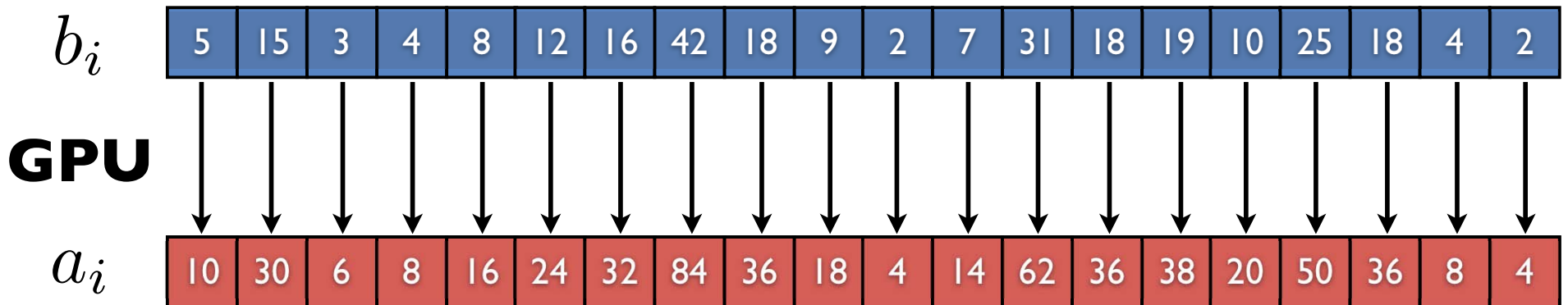
Example:

for i from 1 to N
 $a_i = 2 \cdot b_i$



data-parallel:

$a_i = 2 \cdot b_i$ –run N threads



CUDA tools

- **Compiler:** nvcc
 - C/C++ compiler with a few special directives for programming the GPU
- **Debugger:** cuda-gdb (*linux*) and nexus (*windows*)
 - Full hardware debugger, capable of setting breakpoints and stepping through code
- **Memory checker:** cuda-memcheck (*linux & mac*)
- **Profiler:** cudaprof
 - Accesses numerous performance counters on the GPU
- **CUBLAS:** A full BLAS implementation on the GPU
- **CUFFT:** General purpose FFT library that runs on the GPU (API is like FFTW)

Additional resources to learn CUDA

- **CUDA programming guide** and **best practices guide**:
http://developer.nvidia.com/object/cuda_3_0_downloads.html
Read them cover to cover
- CUDA SDK (a set of good examples of CUDA applications)
 - http://developer.nvidia.com/object/cuda_sdk_samples.html
- The CUDA Forums:
<http://forums.nvidia.com/index.php?showforum=62>
- Course at UIUC:
<http://courses.ece.illinois.edu/ece498/al/>
- Many-core summer school, Virtual School of Computational Science & Engineering:
<http://www.greatlakesconsortium.org>
- Many more links:
http://www.nvidia.com/object/cuda_education.html

Conclusions

- Many-core processors
 - Widely available and inexpensive GPUs
 - Provide 10-100x performance boosts
 - Drastically decrease the time to discovery
 - Architecture differs from standard CPUs
 - Necessitates a new programming model -
more on this in the next lectures