C for CUDA

- First Native C Environment for GPUs
  - 5+ calendar years in development
  - Shipping for over 2 years

- Massive Adoption
  - 25,000+ active developers
  - 100+ applications
  - 30+ NVIDIA GPU clusters using CUDA tool chain

- Feature Rich
  - Available on Windows, Linux, Mac OS (Solaris coming soon)
  - FFT, BLAS, Sparse Matrix, Data Parallel Primitives, LAPACK
  - Matlab, Mathematica, LabView supported by C for CUDA
C for CUDA Roadmap

<table>
<thead>
<tr>
<th>Year</th>
<th>Q1</th>
<th>Q2</th>
<th>Q3</th>
<th>Q4</th>
<th>Q1</th>
<th>Q2</th>
<th>Q3</th>
<th>Q4</th>
<th>Q1</th>
<th>Q2</th>
<th>Q3</th>
<th>Q4</th>
</tr>
</thead>
<tbody>
<tr>
<td>2006</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2007</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2008</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2009</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

CUDA 1.0
CUDA 1.1
CUDA 2.0
CUDA 2.1
CUDA 2.2
CUDA 2.3
CUDA 3.0
CUDA 2.1 (out now!)

- Support for using a GPU that is not driving a display on Vista
- DirectX 10 interoperability (textures, buffers, etc.)
- Visual Studio 2008 Support
- Just-in-time (JIT) PTX compilation
  - For applications that dynamically generate PTX CUDA kernels
- CUDA Debugger beta for 32-bit Linux
- C++ templates are now supported in CUDA kernels
  - This has worked for a while, but now officially supported
- Recent Linux distro support
  - Including Fedora9, OpenSUSE 11 and Ubuntu 8.04
CUDA 2.2 (beta to registered developers now!)

- Zero-copy access to pinned system memory
- Asynchronous memcopy support on Windows Vista
- Texturing from pitch linear memory (i.e. write to texture)
- CUDA Debugger (cudagdb) support for 64-bit Linux
- CUDA OpenGL interop with Texture Objects and FBOs
- Additional counters supported in the CUDA Visual Profiler (cudaprof)
- __threadfence() and __threadfence_block() (memory fences)
CUDA 2.3

- CUDA application profiles
  - allow end-users to configure GPU availability to applications
- CUDA Debugger (cudagdb) support for nested variables
- Query GPU connections to find GPUs not already in use
  - Useful for cluster management
- More…

- Note: subject to change
NVIDIA’s OpenCL Roadmap

<table>
<thead>
<tr>
<th>2008</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>Q2</td>
<td>Q3</td>
<td>Q4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>2009</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>Q2</td>
<td>Q3</td>
<td>Q4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Alpha OpenCL**
- **Beta OpenCL**
- **OpenCL 1.0**
- **OpenCL 1.1**
NVIDIA Professor Partnership

http://www.nvidia.com/page/professor_partnership.html

- Support faculty research & teaching efforts
  - Small equipment gifts (1-2 GPUs)
  - Significant discounts on GPU purchases
    - Especially Quadro, Tesla equipment
    - Useful for cost matching
  - Research contracts
  - Small cash grants (typically ~$25K gifts)
  - Medium-scale equipment donations (10-30 GPUs)

- Informal proposals, reviewed quarterly
  - Focus areas: GPU computing, especially with an educational mission or component

Easy

Competitive

Informal proposals, reviewed quarterly

Focus areas: GPU computing, especially with an educational mission or component