GPU computing: Achievements and perspectives with an emphasis on evolutionary computation

Victoria University of Wellington.
February, 16th, 2015

Manuel Ujaldón
Professor @ Computer Architecture Department, University of Malaga (Spain)
NVIDIA CUDA Fellow
Some reasons why I am here

UMA and VUW are linked in an Erasmus Mundus project:
- There are grants to apply for mobility for post-docs and staff if we find cooperation areas, and evolutionary computation + GPU can be one of them.

Our group has implemented evolutionary computation algorithms on GPUs, with good performance results.
- But I do not know much beyond Ant Colony Optimization (ACO).

I train students on GPU computing as CUDA Fellow sponsored by Nvidia (>40 courses around the world).
- I wanted to show its potential to VUW people.
- Nvidia to donate a Tesla K40 GPU (5000 USD value) to start GPGPU computing in this department if VUW people is willing to play.
Talk outline [40 slides]

1. The GPU evolution [9]
2. Supercomputing on GPUs [5]
3. Implementing applications on GPUs [8]
6. Final remarks on evolutionary computation [8]
I. The GPU evolution
Summary of GPU evolution

- 2001: First many-cores (vertex and pixel processors).
- 2003: Those processor become programmable (with Cg).
- 2006: Vertex and pixel processors unify.
- 2008: Double precision floating-point arithmetic.
- 2010: Operands are IEEE-normalized and memory is ECC.
- 2012: Wider support for irregular computing.
- 2014: The CPU-GPU memory space is unified.

Still pending: Reliability in clusters and connection to disk.
Performance Lead Continues to Grow:

Peak Double Precision FLOPS  Peak Memory Bandwidth

GPU 6x more bandwidth:
- GPU: 3000 GFLOPS
- CPU: 500 GFLOPS
- 7 GHz x 48 bytes = 336 GB/s.
- 2 GHz x 32 bytes = 64 GB/s.
GPUs: More than a platform to play games
The impressive evolution of CUDA

Year 2008

- 100.000.000 CUDA-capable GPUs
- 150.000 CUDA downloads
- 1 supercomputer
- 60 university courses
- 4.000 academic papers

Year 2014

- 500.000.000 CUDA-capable GPUs
- 2.100.000 CUDA downloads
- 75 supercomputers
- 840 university courses
- 40.000 academic papers

The CUDA soft. is downloaded more than once a minute.

Manuel Ujaldón
+800 UNIVERSITIES Teaching CUDA
Worldwide distribution of CUDA university courses
242 popular GPU-accelerated applications
One of the reasons why CUDA is so popular: Much easier than expected

The code to be written in CUDA can be lower than 5%, but exceed 50% of the execution time if remains on CPU.
Three features for the GPU to become a unique processor

- **Simplified control.**
  - The hardware control for a thread is amortized on a warp size of 32 threads. This feature defines the personality for the processor and its affinity with vector and superscalar architectures.

- **Scalability.**
  - Take advantage of the huge data volume handled by applications, to define a sustainable SIMD parallelization model.

- **Productivity.**
  - Lots of mechanisms are defined so that when a thread starts processing slow instructions, others hide its latency taking over resources immediately.
Three reason for feeling attracted to GPUs

Power
Those days of requiring 200 W. are over. Now, GPUs dominate in the Green 500 list. Progression in double precision:

Cost
- Low price due to a massive selling marketplace.
- Three GPUs are sold for each CPU, and the ratio keeps growing.

Ubiquitous
- Everybody already has a bunch of GPUs.
- And you can purchase one almost everywhere.
II. Supercomputing on GPUs
A summary in the evolution of supercomputers

Era of vector computing

Era of distributed computing

Era of accelerators. Heterogeneous computing

Half of the top 10 supercomputers use GPUs, including the first two.

<table>
<thead>
<tr>
<th>RANK</th>
<th>SITE</th>
<th>SYSTEM</th>
<th>CORES</th>
<th>RMAX [TFLOP/S]</th>
<th>RPEAK [TFLOP/S]</th>
<th>POWER [KW]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>National Super Computer Center in Guangzhou China</td>
<td>Tianhe-2 [MilkyWay-2] - TH-IVB-FEP Cluster, Intel Xeon E5-2692 12C 2.200GHz, TH Express-2, Intel Xeon Phi 31S1P NUDT</td>
<td>3,120,000</td>
<td>33,862.7</td>
<td>56,902.4</td>
<td>17,808</td>
</tr>
<tr>
<td>2</td>
<td>DOE/SC/Oak Ridge National Laboratory United States</td>
<td>Titan - Cray XK7, Opteron 6274 16C 2.200GHz, Cray Gemini interconnect, NVIDIA K20x Cray Inc.</td>
<td>560,640</td>
<td>17,590.0</td>
<td>27,112.5</td>
<td>8,209</td>
</tr>
<tr>
<td>3</td>
<td>DOE/NNSA/LLNL United States</td>
<td>Sequoia - BlueGene/Q, Power BQC 16C 1.60 GHz, Custom IBM</td>
<td>1,572,864</td>
<td>17,173.2</td>
<td>20,132.7</td>
<td>7,890</td>
</tr>
<tr>
<td>4</td>
<td>RIKEN Advanced Institute for Computational Science (AICS) Japan</td>
<td>K computer, SPARC64 VIIIfx 2.0GHz, Tofu interconnect Fujitsu</td>
<td>705,024</td>
<td>10,510.0</td>
<td>11,280.4</td>
<td>12,660</td>
</tr>
<tr>
<td>5</td>
<td>DOE/SC/Argonne National Laboratory United States</td>
<td>Mira - BlueGene/Q, Power BQC 16C 1.60GHz, Custom IBM</td>
<td>786,432</td>
<td>8,586.6</td>
<td>10,066.3</td>
<td>3,945</td>
</tr>
<tr>
<td>6</td>
<td>Swiss National Supercomputing Centre (CSCS) Switzerland</td>
<td>Piz Daint - Cray XC30, Xeon E5-2670 8C 2.600GHz, Aries interconnect, NVIDIA K20x Cray Inc.</td>
<td>115,984</td>
<td>6,271.0</td>
<td>7,788.9</td>
<td>2,325</td>
</tr>
<tr>
<td>7</td>
<td>Texas Advanced Computing Center/Univ. of Texas United States</td>
<td>Stampede - PowerEdge C8220, Xeon E5-2680 2C 2.700GHz, Infiniband FDR, Intel Xeon Phi SE10P Dell</td>
<td>462,442</td>
<td>5,168.1</td>
<td>8,520.1</td>
<td>4,510</td>
</tr>
<tr>
<td>8</td>
<td>Forschungszentrum Juelich (FZJ) Germany</td>
<td>JUQUEEN - BlueGene/Q, Power BQC 16C 1.600GHz, Custom Interconnect IBM</td>
<td>458,752</td>
<td>5,008.9</td>
<td>5,872.0</td>
<td>2,301</td>
</tr>
<tr>
<td>9</td>
<td>DOE/NNSA/LLNL United States</td>
<td>Vulcan - BlueGene/Q, Power BQC 16C 1.600GHz, Custom Interconnect IBM</td>
<td>393,216</td>
<td>4,293.3</td>
<td>5,033.2</td>
<td>1,972</td>
</tr>
<tr>
<td>10</td>
<td>Government United States</td>
<td>Cray CS-Storm, Intel Xeon E5-2660v2 10C 2.2GHz, Infiniband FDR, Nvidia K40 Cray Inc.</td>
<td>72,800</td>
<td>3,577.0</td>
<td>6,131.8</td>
<td>1,499</td>
</tr>
</tbody>
</table>
Graphics accelerators in the Top500.org
All supercomputers in green500.org between positions 3\textsuperscript{rd} and 18\textsuperscript{th} have Nvidia GPUs

<table>
<thead>
<tr>
<th>Green500 Rank</th>
<th>MFLOPS/W</th>
<th>Site*</th>
<th>Computer*</th>
<th>Total Power (kW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5,271.81</td>
<td>GSI Helmholtz Center</td>
<td>L-CSC - ASUS ESC4000 FDR/G2S, Intel Xeon E5-2690v2 10C 3GHz, Infiniband FDR, AMD FirePro S9150 Level 1 measurement data available</td>
<td>57.15</td>
</tr>
<tr>
<td>2</td>
<td>4,945.63</td>
<td>High Energy Accelerator Research Organization /KEK</td>
<td>Suiren - ExaScaler 32U256SC Cluster, Intel Xeon E5-2660v2 10C 2.2GHz, Infiniband FDR, PEZY-SC</td>
<td>37.83</td>
</tr>
<tr>
<td>3</td>
<td>4,447.58</td>
<td>GSIC Center, Tokyo Institute of Technology</td>
<td>TSUBAME-KFC - LX 1U-4GPU/104Re-1G Cluster, Intel Xeon E5-2620v2 6C 2.100GHz, Infiniband FDR, NVIDIA K20x</td>
<td>35.39</td>
</tr>
<tr>
<td>4</td>
<td>3,962.73</td>
<td>Cray Inc.</td>
<td>Storm1 - Cray CS-Storm, Intel Xeon E5-2660v2 10C 2.2GHz, Infiniband FDR, NVIDIA K40m Level 3 measurement data available</td>
<td>44.54</td>
</tr>
<tr>
<td>5</td>
<td>3,631.70</td>
<td>Cambridge University</td>
<td>Wilkes - Dell T620 Cluster, Intel Xeon E5-2630v2 6C 2.600GHz, Infiniband FDR, NVIDIA K20</td>
<td>52.62</td>
</tr>
<tr>
<td>6</td>
<td>3,543.32</td>
<td>Financial Institution</td>
<td>iDataPlex DX360M4, Intel Xeon E5-2680v2 10C 2.800GHz, Infiniband, NVIDIA K20x</td>
<td>54.60</td>
</tr>
<tr>
<td>7</td>
<td>3,517.84</td>
<td>Center for Computational Sciences, University of Tsukuba</td>
<td>HA-PACS TCA - Cray CS300 Cluster, Intel Xeon E5-2680v2 10C 2.800GHz, Infiniband QDR, NVIDIA K20x</td>
<td>78.77</td>
</tr>
<tr>
<td>8</td>
<td>3,459.46</td>
<td>SURFsara</td>
<td>Cartesius Accelerator Island - Bullx B515 cluster, Intel Xeon E5-2450v2 8C 2.5GHz, InfiniBand 4x FDR, NVIDIA K40m</td>
<td>44.40</td>
</tr>
<tr>
<td>9</td>
<td>3,185.91</td>
<td>Swiss National Supercomputing Centre (CSCS)</td>
<td>Piz Daint - Cray XC30, Xeon E5-2670 8C 2.600GHz, Aries interconnect, NVIDIA K20x Level 3 measurement data available</td>
<td>1,753.66</td>
</tr>
<tr>
<td>10</td>
<td>3,131.06</td>
<td>ROMEO HPC Center - Champagne-Ardenne</td>
<td>romeo - Bull R421-E3 Cluster, Intel Xeon E5-2650v2 8C 2.600GHz, Infiniband FDR, NVIDIA K20x</td>
<td>81.41</td>
</tr>
</tbody>
</table>

Manuel Ujaldón
Building a GPU-based supercomputer:
Titan [Cray XK7] as case study

The chips
A 16-core CPU plus a 2688-core GPU

130 GFLOPS + 1310 GFLOPS (peak on double-precision floating-point format)
III. Implementing applications on GPUs
There is a platform for each user profile
... even for students
The CUDA family picture

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>GeForce 600 Series</td>
<td>Quadro Kepler Series</td>
<td>Quadro Fermi Series</td>
<td>QuadroFX Series</td>
</tr>
<tr>
<td>Quadro Kepler Series</td>
<td>GeForce 500 Series</td>
<td>GeForce 400 Series</td>
<td>Quadro Plex Series</td>
</tr>
<tr>
<td>Tesla K20 Tesla K10</td>
<td>GeForce 200 Series</td>
<td>GeForce 9 Series</td>
<td>Quadro NVS Series</td>
</tr>
<tr>
<td></td>
<td>GeForce 8 Series</td>
<td>GeForce 8 Series</td>
<td>Tesla 10 Series</td>
</tr>
</tbody>
</table>

Programming Languages

- C
- C++
- Fortran
- Java
- Python Wrappers
- DirectCompute
- Directives (e.g., OpenACC)

CUDA Family Picture

- GeForce
- Quadro
- Tesla

Libraries and Middleware

- CUFFT
- CUBLAS
- CURAND
- CUSPARSE
- CUDA
- MAGMA
- Thrust
- NPP
- VSIPL
- SVM
- OptiX
- OpenCL
- Ray
- MATLAB
- Mathematica

GPU Computing Applications
Choose processor depending on application: And evolutionary computation is in between

Massively parallel computing

Control and communication

Productivity-based applications

Data intensive applications

CPU
(Sequential computing)

GPU
(Parallel computing)

Graphics

Oil & Gas  Finance  Medical  Biophysics  Numerics  Audio  Video  Imaging
Acceleration factors depending of the area of application

Particle systems
Simulation in physics
Molecular dynamics

Typical factor gains on GPU: 2-3x

Signal processing
Volume rendering
Image processing
Bioinformatics

Typical factor gains on GPU: 10-20x

Evolutionary computation
Database queries
Data mining
Reduction operators

Typical factor gains on GPU: 5-10x

Ray tracing
Visualization
Photon mapping

GPU gains: > 20x
Scalability on grand-challenge problems: Much better using GPUs

Without GPU: Weak scalability
With GPU: Strong scalability
Mapping hardware and software concepts

GPU

Multiprocessor N

Multiprocessor 2

Multiprocessor 1

Shared memory

Registers

Processor 1

Processor 2

...

Processor M

Control Unit (SIMD)

Constant cache

Texture cache

Global memory

Thread

per-Thread Private Local Memory

Thread Block

per-Block Shared Memory

Grid 0

per-Application Context Global Memory

Grid 1
### Scalability for the architecture: A summary of 3 generations

<table>
<thead>
<tr>
<th>Architecture</th>
<th>Tesla</th>
<th>Fermi</th>
<th>Kepler</th>
<th>CUDA Compute Capability (CCC)</th>
<th>N (multiprocs.)</th>
<th>M (cores/multip.)</th>
<th>Number of cores</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>G80</td>
<td>GT200</td>
<td>GF100</td>
<td>GF104 (K10)</td>
<td>16</td>
<td>8</td>
<td>128</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>GK110 (K20)</td>
<td>30</td>
<td>8</td>
<td>240</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>GK110 (K40)</td>
<td>16</td>
<td>32</td>
<td>512</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2x GK210 (K80)</td>
<td>7</td>
<td>48</td>
<td>336</td>
</tr>
<tr>
<td>Time frame</td>
<td>2006-07</td>
<td>2008-09</td>
<td>2010</td>
<td>2011</td>
<td>8</td>
<td>192</td>
<td>1536</td>
</tr>
<tr>
<td></td>
<td>2012</td>
<td>2013</td>
<td>2013-14</td>
<td>2014</td>
<td>13</td>
<td>192</td>
<td>2496</td>
</tr>
<tr>
<td></td>
<td>2013-14</td>
<td>2014</td>
<td></td>
<td></td>
<td>15</td>
<td>192</td>
<td>2880</td>
</tr>
</tbody>
</table>

Manuel Ujaldón
A typical GPU: Kepler
IV. A look to the future:
Power and memory
A 2015 graphics card: Kepler GPU with GDDR5 video memory
A look ahead through Nvidia’s GPU roadmap

GFLOPS in double precision for each watt consumed

- Tesla (CUDA)
- Fermi (FP64)
- Maxwell
- Pascal

2008 - 2016

Feature Highlight:
- Dynamic Parallelism
- Unified memory
- 3D Memory
- NVLink

Manuel Ujaldón
A 2016/17 graphics card:
Pascal GPU with Stacked (3D) DRAM
Micron’s prototypes (upper) and Viking’s commercial products (lower)

BGA Stacking: Enabling High-Capacity DRAM Modules

BGA Stacking - enabling the highest capacity memory modules available in the industry. Learn more.
Nvidia’s development announcement [GTC’14]

- 3D chip-on-wafer integration.
- Many X bandwidth.
- 2.5x capacity.
- 4x energy efficiency.
Unified memory: The idea

- CPU
- GPU
- DDR3
- GDDR5
- Main memory
- Video memory

- Dual-, tri- or quad-channel (~100 GB/s.)
- PCI-express (~10 GB/s.)
- 256, 320, 384 bits (~300 GB/s.)

- Already available in Maxwell GPUs since 2014.
V. Programming alternatives
CUDA provides a rich toolchain & ecosystem for fast ramp-up on GPUs

- **Numerical Packages**
  - MATLAB
  - Mathematica
  - NI LabView
  - pyCUDA

- **Debuggers & Profilers**
  - cuda-gdb
  - Visual Profiler
  - Parallel Nsight
  - Visual Studio
  - Allinea
  - TotalView

- **GPU Compilers**
  - C
  - C++
  - Fortran
  - OpenCL
  - DirectCompute
  - Java
  - Python

- **Parallelizing Compilers**
  - PGI Accelerator
  - CAPS
  - HMPP
  - mCUDA
  - OpenMP

- **Libraries**
  - BLAS
  - FFT
  - LAPACK
  - NPP
  - Sparse
  - Imaging
  - RNG

- **GPGPU Consultants & Training**
  - ANEO
  - GPU Tech

- **OEM Solution Providers**
  - Microsoft

- **Proprietary Software & Hardware**
  - DELL
  - IBM
  - Cray
  - ASUS
  - SUPERMICRO
  - sgi
  - Fujitsu
  - Bull
  - APPs
  - lenovo
  - NEC
CUDA Parallel Computing Platform

Programming Approaches

- Libraries
  - “Drop-in” Acceleration
- OpenACC Directives
  - Easily Accelerate Apps
- Programming Languages
  - Maximum Flexibility

Development Environment

- Nsight IDE
  - Linux, Mac and Windows
  - GPU Debugging and Profiling
- CUDA-GDB debugger
  - NVIDIA Visual Profiler

Open Compiler Tool Chain

- Enables compiling new languages to CUDA platform, and CUDA languages to other architectures

Hardware Capabilities

- SMX
- Dynamic Parallelism
- HyperQ
- GPUDirect

Enables compiling new languages to CUDA platform, and CUDA languages to other architectures
### GPU accelerated libraries: “Drop-in” acceleration for your applications

<table>
<thead>
<tr>
<th>Linear Algebra</th>
<th>Numerical &amp; Math</th>
<th>Data Struct. &amp; AI</th>
<th>Visual Processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>FFT, BLAS, SPARSE, Matrix</td>
<td>NVIDIA Math Lib</td>
<td>Sort, Scan, Zero Sum</td>
<td>NVIDIA NPP</td>
</tr>
<tr>
<td>NVIDIA cuFFT, cuBLAS, cuSPARSE</td>
<td>ArrayFire</td>
<td></td>
<td>NVIDIA Video Encode</td>
</tr>
<tr>
<td>CULA tools</td>
<td>NVIDIA cuBLAS, cuSPARSE</td>
<td>Thrust</td>
<td></td>
</tr>
<tr>
<td>MAGMA</td>
<td>NVIDIA Math Lib</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CUSP</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Manuel Ujaldón
OpenACC: Open, Simple, Portable

main() {
  ...
  <serial code>
  ...
  #pragma acc kernels
  {
    <compute intensive code>
  }
  ...
  
}

- Open Standard
- Easy, Compiler-Driven Approach
- Portable on GPUs and Xeon Phi

CAM-SE Climate
6x Faster on GPU
Top Kernel: 50% of Runtime

Available from:

PGI CAPS CRAY

Manuel Ujaldón
VI. Remarks on evolutionary computation
## Platforms to compare

<table>
<thead>
<tr>
<th>Vendor</th>
<th>Microarchitecture</th>
<th>Model</th>
<th>GB/s.</th>
<th>GFLOP/s.</th>
<th>Byte/FLOP</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMD</td>
<td>Bulldozer</td>
<td>Opteron 6284</td>
<td>59,7</td>
<td>217,6 (DP)</td>
<td>0,235</td>
</tr>
<tr>
<td>AMD</td>
<td>Souther Islands</td>
<td>Radeon HD7970</td>
<td>288</td>
<td>1010 (DP)</td>
<td>0,285</td>
</tr>
<tr>
<td>Intel</td>
<td>Sandy Bridge</td>
<td>Xeon E5-2690</td>
<td>51,2</td>
<td>243,2 (DP)</td>
<td>0,211</td>
</tr>
<tr>
<td>Intel</td>
<td>MIC</td>
<td>Xeon Phi</td>
<td>300</td>
<td>1024 (DP)</td>
<td>0,292</td>
</tr>
<tr>
<td>Nvidia</td>
<td>Fermi GF110</td>
<td>Tesla M2090 (16 SMs)</td>
<td>177</td>
<td>665 (DP)</td>
<td>0,266</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1331 (SP)</td>
<td>0,133</td>
</tr>
<tr>
<td>Nvidia</td>
<td>Kepler GK110</td>
<td>Tesla K20X (14 SMXs)</td>
<td>250</td>
<td>1310 (DP)</td>
<td>0,190</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3950 (SP)</td>
<td>0,063</td>
</tr>
<tr>
<td>Nvidia</td>
<td>Pascal GPU</td>
<td>with Stacked 3D DRAM</td>
<td>1024</td>
<td>4000 (DP)</td>
<td>0,256</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12000 (SP)</td>
<td>0,085</td>
</tr>
</tbody>
</table>

FLOP/byte (operational intensity) = GFLOP/s / GB/s
The Roofline model: Hardware vs. Software

[Graph showing the relationship between GFLOP/s (double precision performance) and FLOP/byte (operational intensity) with different hardware and kernels.

<table>
<thead>
<tr>
<th>Processor</th>
<th>GB/s.</th>
<th>GFLOP/s.</th>
<th>B/FLOP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Opteron</td>
<td>60</td>
<td>217 (DP)</td>
<td>0.235</td>
</tr>
<tr>
<td>Radeon</td>
<td>288</td>
<td>1010 (DP)</td>
<td>0.285</td>
</tr>
<tr>
<td>Xeon</td>
<td>51</td>
<td>243 (DP)</td>
<td>0.211</td>
</tr>
<tr>
<td>Xeon Phi</td>
<td>300</td>
<td>1024 (DP)</td>
<td>0.292</td>
</tr>
<tr>
<td>Fermi</td>
<td>177</td>
<td>665 (DP)</td>
<td>0.266</td>
</tr>
<tr>
<td>Kepler</td>
<td>250</td>
<td>1310 (DP)</td>
<td>0.190</td>
</tr>
<tr>
<td>Pascal</td>
<td>1024</td>
<td>4000 (DP)</td>
<td>0.256</td>
</tr>
</tbody>
</table>
The Roofline model: Software evolution. Case study: Evolutionary computation

[Graph showing the Roofline model with performance data for Pascal and Kepler architectures. Milestones include early adopters in 2006, our ACO implementation in 2010, recent enhancements in 2014, and expected in 2018.]
Our work on evolutionary computation

We developed a framework for implementing swarm intelligence algorithms: Ant colony, bird flocking, bacterial growth, ...

We focus on massively parallel methods inspired by nature whose simulation in sequential or modestly parallel architectures compromise their effectiveness.

Major challenges:

- Different levels of massive parallelism.
- Novel synchronization and communication patterns.
- Memory bounded problems.
Our contribution to the ACO parallelization

Two main stages for parallelization purposes:
- Tour Construction and Pheromone Update.

Task parallelism in **Tour Construction** does not fit well on GPUs. We proposed a new data-parallel approach, to attain over **20x speed-up** compared to a similar CPU code.

We implemented the first **Pheromone Update** on GPUs, identifying potential tradeoffs and investigating different alternatives to sustain gains over **20x** as well.

We validated the results vs. solution computed on CPUs.
# List of publications: 6 journals and 8 conferences

## International Journals


List of publications (2)

National & International Conferences


List of publications (3)

**National & International Conferences**


Thanks for your attention!

You can always reach me in Spain at the Computer Architecture Department of the University of Malaga:

- e-mail: ujaldon@uma.es
- Phone: +34 952 13 28 24.

Or, more specifically on GPUs, visit my web page as Nvidia CUDA Fellow:

http://research.nvidia.com/users/manuel-ujaldon
Acknowledgements to Nvidia
VII. Bibliography and tools
CUDA Zone: Basic web resource for a CUDA programmer

[developer.nvidia.com/cuda-zone]
CUDA 6 Production Release.
Free download for all platforms and users

[developer.nvidia.com/cuda-downloads]
CUDA books: From 2007 to 2013

**GPU Gems series** [developer.vidia.com/content/GPUGems3/gpugems3_part01.html]

**List of CUDA books in** [www.nvidia.com/object/cuda_books.html]
Guides for developers and more documents

Getting started with CUDA C: Programmers guide.
[docs.nvidia.com/cuda/cuda-c-programming-guide]

For tough programmers: The best practices guide.
[docs.nvidia.com/cuda/cuda-c-best-practices-guide]

The root web collecting all CUDA-related documents:
[docs.nvidia.com/cuda]

where we can find, additional guides for:

- Installing CUDA on Linux, MacOS and Windows.
- Optimize and improve CUDA programs on Kepler platforms.
- Check the CUDA API syntax (runtime, driver and math).
- Learn to use libraries like cuBLAS, cuFFT, cuRAND, cuSPARSE, ...
- Deal with basic tools (compiler, debugger, profiler).
Choices to accelerate your applications on GPUs and material for teaching CUDA

[developer.nvidia.com/cuda-education-training] (also available from the left lower corner of the CUDA Zone)

CUDA Education & Training

Accelerate Your Applications
Learn using step-by-step instructions, video tutorials and code samples.
- Accelerated Computing with C/C++
- Accelerate Applications on GPUs with OpenACC Directives
- Accelerated Numerical Analysis Tools with GPUs
- Drop-in Acceleration on GPUs with Libraries
- GPU Accelerated Computing with Python

Teaching Resources
Get the latest educational slides, hands-on exercises and access to GPUs for your parallel programming courses.
- Parallel Programming Training Materials
- NVIDIA Research & Academic Programs

Sign up to join the Accelerated Computing Educators Network. This network seeks to provide a collaborative area for those looking to educate others on massively parallel programming. Receive updates on new educational material, access to CUDA Cloud Training Platforms, special events for educators, and an educators focused news letter.

Sign-up Today!
Courses on-line (free access)

More than 50,000 registered users from 127 countries over the last 6 months. An opportunity to learn from CUDA masters:

- Prof. Wen-Mei Hwu (Univ. of Illinois).
- Prof. John Owens (Univ. of California at Davis).
- Dr. David Luebke (Nvidia Research).

There are two basic options, both recommended:

- Introduction to parallel programming:
  - 7 units of 3 hours = 21 hours.
  - Provides high-end GPUs to carry out the proposed assignments.
  - [https://developer.nvidia.com/udacity-cs344-intro-parallel-programming]

- Heterogeneous Parallel Programming: [coursera]
  - 9 weeks, each with classes (20’ video), quizzes and programming assignments.
  - [https://www.coursera.org/course/hetero]
Tutorials about C/C++, Fortran and Python

You have to register on the Amazon EC2 services available on the Web (cloud computing): [nvidia.qwiklab.com]

They are usually sessions of 90 minutes.

Only a Web browser and SSH client are required.

Some tutorials are free, other require tokens of $29.99.
Talks and webinars

Talks recorded at GTC (Graphics Technology Conference):
- 383 talks from 2013.
- More than 500 available from 2014.
- [www.gputechconf.com/gtcnew/on-demand-gtc.php]

Webinars about GPU computing:
- List of past talks on video (mp4/wmv) and slides (PDF).
- List of incoming on-line talks to be enrolled.
- [developer.nvidia.com/gpu-computing-webinars]

CUDACasts:
- [bit.ly/cudacasts]
# Examples of webinars about CUDA 6.0

## GTC Express Webinar Program

GTC Express is a year-round extension of the great content available at GTC. Each month, top developers, scientists, researchers, and industry practitioners present innovative and thought-provoking webinars focused on the GPU-enabled work they're doing and sharing how GPUs are transforming their fields.

Register below for upcoming webinars and explore recordings of previous webinars.

## GTC EXPRESS SCHEDULE AT-A-GLANCE

<table>
<thead>
<tr>
<th>Date</th>
<th>Title</th>
<th>Speaker</th>
<th>Register Now</th>
</tr>
</thead>
<tbody>
<tr>
<td>June 3, 2014, 9:00 AM PDT</td>
<td>The Next Steps for Folding@home</td>
<td>Vijay Pande, Professor, Stanford University</td>
<td><a href="#">Register Now</a></td>
</tr>
<tr>
<td>May 14, 2014, 10:00 AM PDT</td>
<td>CUDA 6: Performance Overview</td>
<td>Jonathan Cohen, Senior Manager, CUDA Libraries and Algorithms, NVIDIA</td>
<td><a href="#">Register Now</a></td>
</tr>
<tr>
<td>May 13, 2014, 9:00 AM PDT</td>
<td>An Overview of AMBER 14 - Creating the World's Fastest Molecular Dynamics Software Package</td>
<td>Ross C. Walker, University of California San Diego, Scott Le Grand, Amazon Web Services, and Adrian Rohtberg, University of Florida.</td>
<td><a href="#">Register Now</a></td>
</tr>
<tr>
<td>May 7, 2014, 10:00 AM PDT</td>
<td>CUDA 6: Drop-in Performance Optimized Libraries</td>
<td>NVIDIA DevTech Team</td>
<td><a href="#">Register Now</a></td>
</tr>
<tr>
<td>May 1, 2014, 10:00 AM PDT</td>
<td>CUDA 6: Unified Memory</td>
<td>Mark Ebersole, CUDA Educator, NVIDIA</td>
<td><a href="#">Register Now</a></td>
</tr>
<tr>
<td>April 23, 2014, 11:00 AM PDT</td>
<td>CUDA 6 Features Overview</td>
<td>Ujval Kapasi, CUDA Product Manager, NVIDIA</td>
<td><a href="#">Register Now</a></td>
</tr>
</tbody>
</table>
Developers

- Sign up as a registered developer:
  - [www.nvidia.com/paralleldeveloper](www.nvidia.com/paralleldeveloper)
  - Access to exclusive developer downloads.
  - Exclusive access to pre-release CUDA installers like CUDA 6.0.
  - Exclusive activities and special offers.

- Meeting point with many other developers:
  - [www.gpucomputing.net](www.gpucomputing.net)

- GPU news and events:
  - [www.gpgpu.org](www.gpgpu.org)

- Technical questions on-line:
  - NVIDIA Developer Forums: [devtalk.nvidia.com](devtalk.nvidia.com)
  - Search or ask on: [stackoverflow.com/tags/cuda](stackoverflow.com/tags/cuda)
Developers (2)

List of CUDA-enabled GPUs:
[developer.nvidia.com/cuda-gpus]

And a last tool for tuning code: The CUDA Occupancy Calculator
[developer.download.nvidia.com/compute/cuda/CUDA_Occupancy_calculator.xls]
Future developments

Nvidia’s blog contains articles unveiling future technology to be used within CUDA. It is the most reliable source about what’s next (subscription recommended):

[devblogs.nvidia.com/parallelforall]

Some recommended articles:

“5 Powerful New Features in CUDA 6”, by Mark Harris.
“CUDA 6.0 Unified Memory”, by Mark Ebersole.
“CUDA Dynamic Parallelism API and Principles”, by Andrew Adinetz.
“NVLINK, Pascal and Stacked Memory: Feeding the Appetite for Big Data”, by Denis Foley.
“CUDA Pro Tip: Increase Application Performance with NVIDIA GPU Boost”, by Mark Harris.