NVIDIA CUDA RESEARCH CENTER

APLICACIONES Y OPORTUNIDADES

Miguel Ángel Martínez del Amor

Research Group on Natural Computing (RGNC)

Universidad de Sevilla

Contenido

- Presentación del NVIDIA CRC
  - Beneficios
  - El RGNC
- Introducción a GPU computing
  - Computación Paralela
  - Evolución de la GPU
  - Introducción a CUDA
  - Otros estándares y tecnologías
- Recursos y oportunidades

Presentación del NVIDIA CRC

NVIDIA CUDA Research Center

- Nominación concedida por la compañía NVIDIA corp.
- Para la Universidad de Sevilla, hasta Enero de 2015.
- Por la “visión, calidad e impacto” de la investigación desarrollada empleando CUDA.

NVIDIA CUDA Research Center

- Beneficios para la comunidad de nuestra universidad:
  - Donación de una Tesla K40.
  - Participación en el Centers Reward Program para descuentos en equipos.
  - Sesiones de formación online.
  - Designación de personal técnico de NVIDIA.
  - 25% dto. para registro en el GTC2015.

NVIDIA CUDA Research Center

Research Group on Natural Computing

- Director: **Mario J. Pérez-Jiménez** (miembro *Academia Europaea*).
- Reconocido por el PAI: TIC-193
- Miembro del European Molecular Consortium
- 12 miembros: 7 matemáticos, 5 ingenieros informáticos

RGNC

Research Group on Natural Computing

- **Natural Computing**: "un área interdisciplinar concerniente a la relación entre Computación y Biología"[1].
- Áreas de investigación:

Computación inspirada en Biología	Biología motivada en Computación	Computación con Biología
• Neural Networks • Genetic Algorithms • Ant Colony Optimizations	• Bioinformática • Biología de Sistemas • Biología sintética	• Quantum Computing • DNA Computing • Membrane Computing[2]

[1] **J. Brownlee**. Clever Algorithms: Nature-Inspired Programming Recipes.

[2] **G. Paun**. Computing with membranes, *JCSS,* 61, 1 (2000), 108-143. http://ppage.psystems.eu

Research Group on Natural Computing

- **Línea de investigación:** Desarrollo de tecnologías habilitadoras basadas en métodos formales bio-inspirados para la especificaciones, simulación, análisis y estudio teórico de fenómenos biológicos.

Modelos bio-inspirados de computación

Aplicación de nuevos paradigmas computacionales inspirados en la Naturaleza viva para el establecimiento de nuevas fronteras de la eficiencia. Caracterización de la conjetura P≠NP en estos modelos no convencionales de computación.

Biología de Sistemas computacional

Aplicación de modelos bio-inspirados a la modelización de sistemas celulares, como rutas señalizadoras involucradas en la proliferación no controlada de células tumorosas, y en la comunicación entre bacterias (p.ej. quorum sensing).

Modelización Ecológica

Desarrollo de modelos probabilísticos y multi-compartimentales de ecosistemas reales basados en modelos de computación bio-inspirados. Desarrollo de herramientas software que permitan a los ecólogos utilizar nuestros modelos.

Computación de Alto Rendimiento con GPUs

Desarrollo de herramientas de simulación de alto rendimiento para modelos bio-inspirados empleando arquitecturas paralelas como la GPU y CPUs multi-núcleo. Administración de un cluster de GPUs en el grupo.

Introducción a GPU Computing

Computación Paralela

Computación Paralela

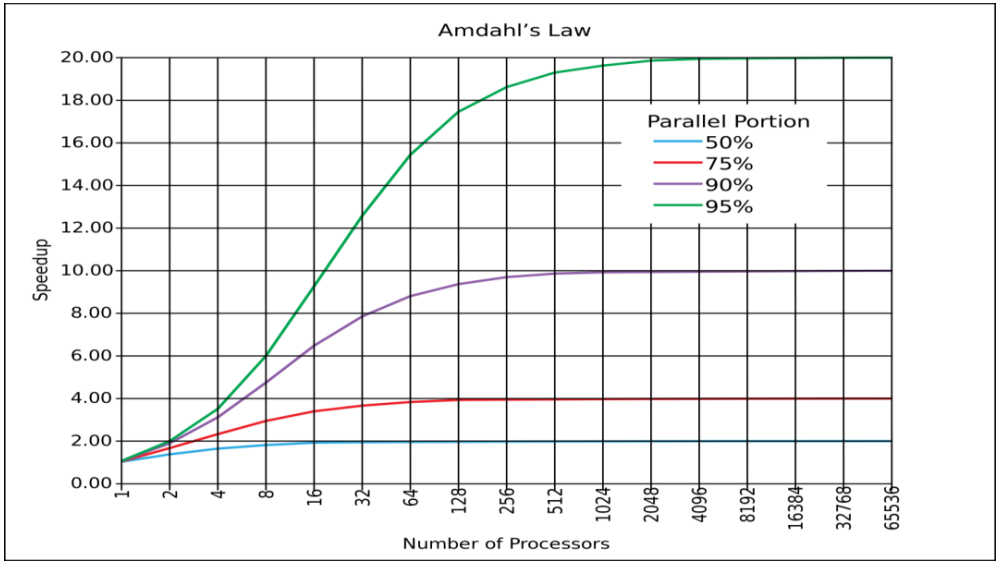
- Formas de acelerar una aplicación:
  - Disminuyendo la ***complejidad*** del algoritmo.
  - Aumentando la velocidad y capacidad del medio de computación: ***frecuencia*** de reloj.
  - Buscando tareas dentro de la aplicación que puedan realizarse de forma ***paralela***.

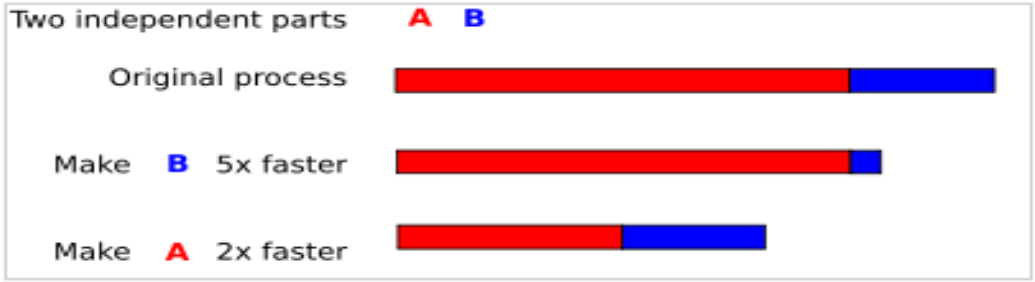
Paralelización de las aplicaciones

- La siguiente vía para acelerar la ejecución, vendría dada por la ***paralelización de*** ciertas ***tareas*** de nuestro programa, de forma que se ***ejecuten de forma simultánea*** (***hilos***).
- Para ello se necesita:
  - **Adaptar la aplicación** para ser ejecutada en paralelo.
  - Disponer de **plataformas paralelas** donde ejecutarse.

Computación paralela

- Ley de Amdahl





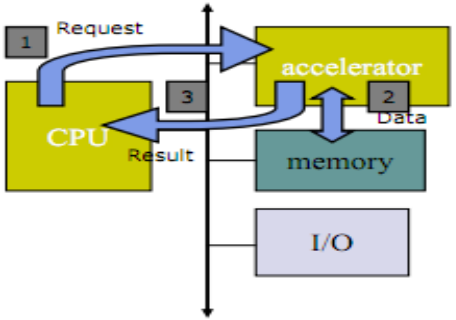
Tipos de paralelismo

Paralelismo de Tareas

Paralelismo de Datos

Aceleradores para HPC

- Dispositivos hardware de propósito específico o general para aumentar el rendimiento de las aplicaciones complementando a la CPU
- Los aceleradores HW están diseñados para código software computacionalmente intensivo



Aceleradores para HPC

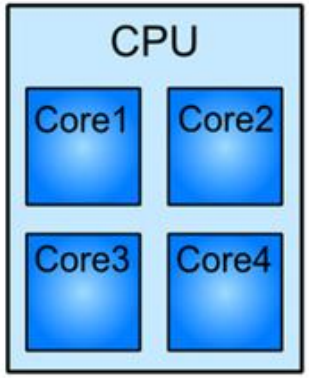
- Ventajas
  - Arquitecturas altamente paralelas y sin caída de nodos
  - Económicas (en comparación con soluciones anteriores)
  - Menor consumo energético y mejor mantenimiento
- Desventajas
  - Programación compleja, y dependiente de arquitectura
  - Capacidad de cómputo limitada a los recursos
- Algunas soluciones:
  - CMP (chip multiprocessor)
  - Cell BE
  - FPGA
  - Intel Phi
  - GPU
  - Híbridos

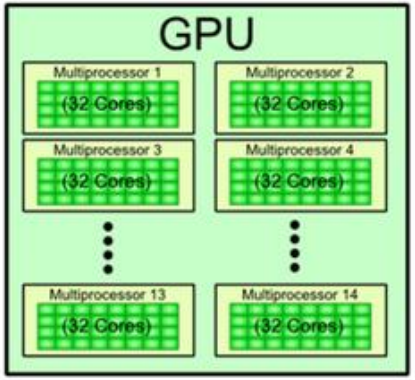
Introducción a GPU Computing

Evolución de la GPU

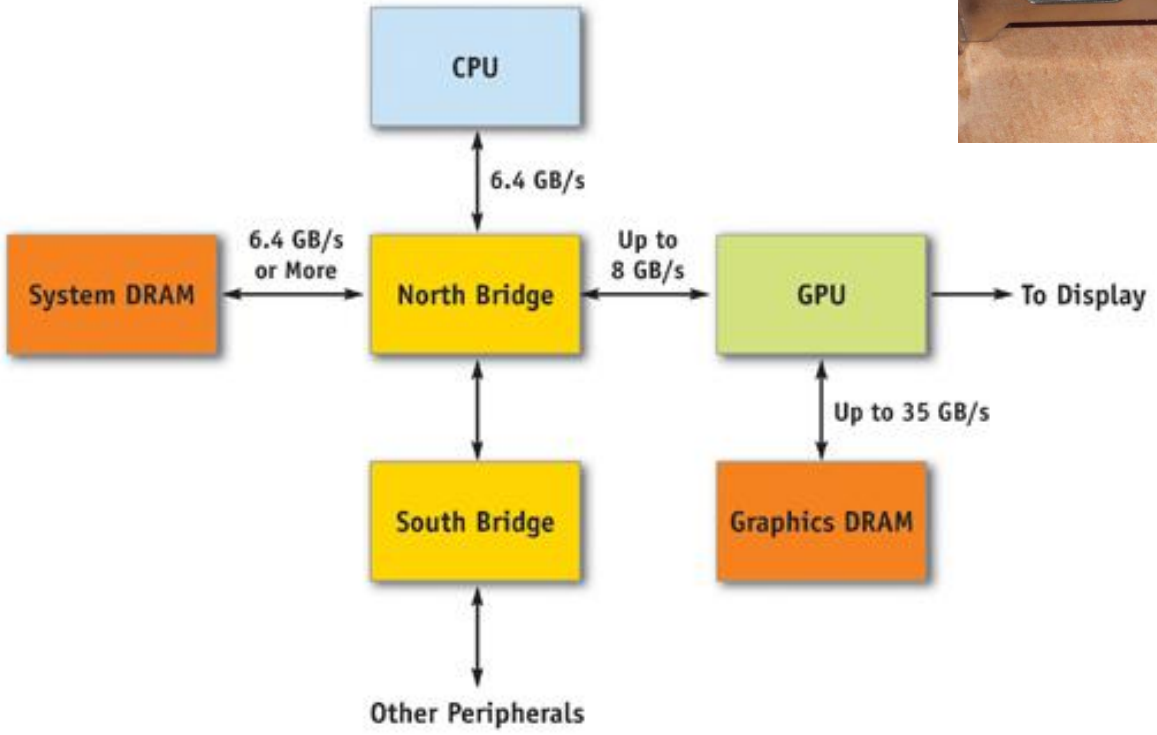
¿Qué es la GPU?

- Graphics Processing Unit.
- El procesador de las **tarjetas gráficas** era de propósito sólo gráfico.
- **GPGPU**: Técnicas de programación para usar la GPU como un co-procesador paralelo.
- Actualidad: Programación intuitiva y alta cantidad de recursos (hasta 2880 núcleos en K40).

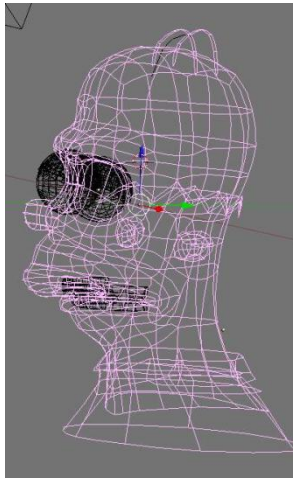


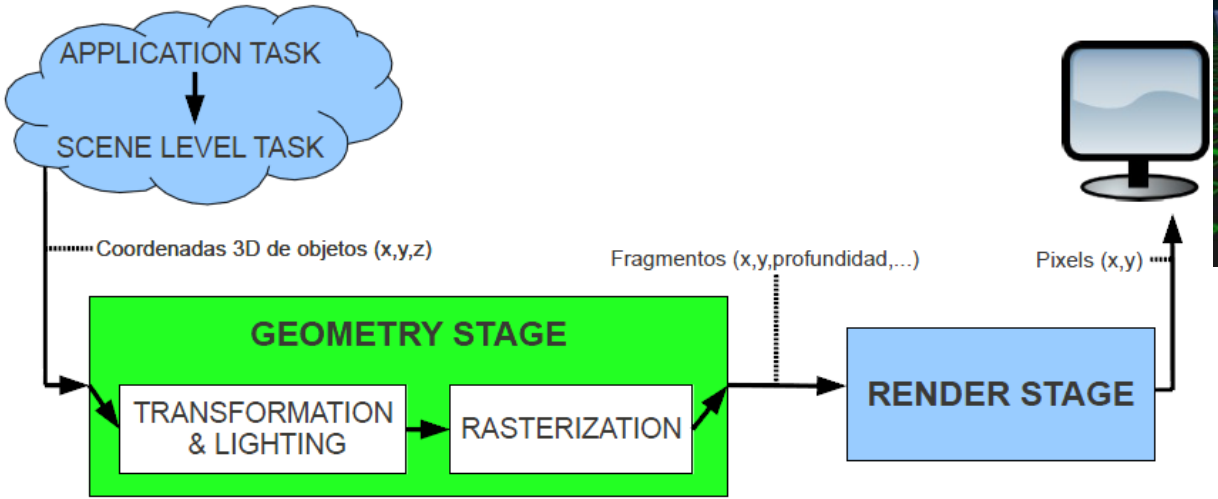


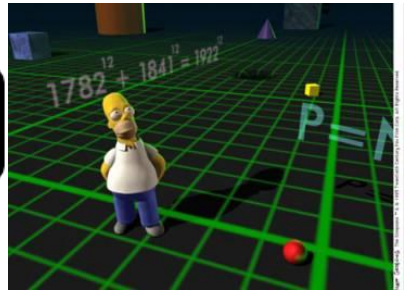
Posición de la GPU



Evolución del pipeline gráfico



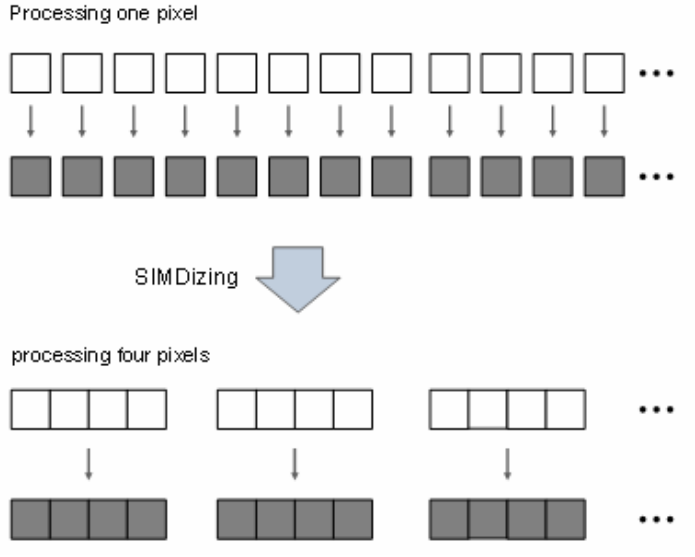




Move Object and Camera (AI, Realistic Physics)	CPU	CPU	CPU	CPU	CPU	CPU
Scene Level Calculations, High order surface tessellation	CPU	CPU	CPU	CPU	CPU	GPU
Transforms	CPU	CPU	CPU	GPU	GPU	GPU
Lighting	CPU	CPU	CPU	GPU	GPU	GPU
Triangle Setup and Clipping	CPU	Graphics Processor	Graphics Processor	GPU	GPU	GPU
Rendering	Graphics Processor	Graphics Processor	Graphics Processor	GPU	GPU	GPU
	1996	1997	1998	1999	2000	2002+

La GPU: Procesadores SIMD

- Paralelismo masivo de **datos**:
  - Gran cantidad de hilos procesando de forma simultánea la misma instrucción sobre distintos datos.

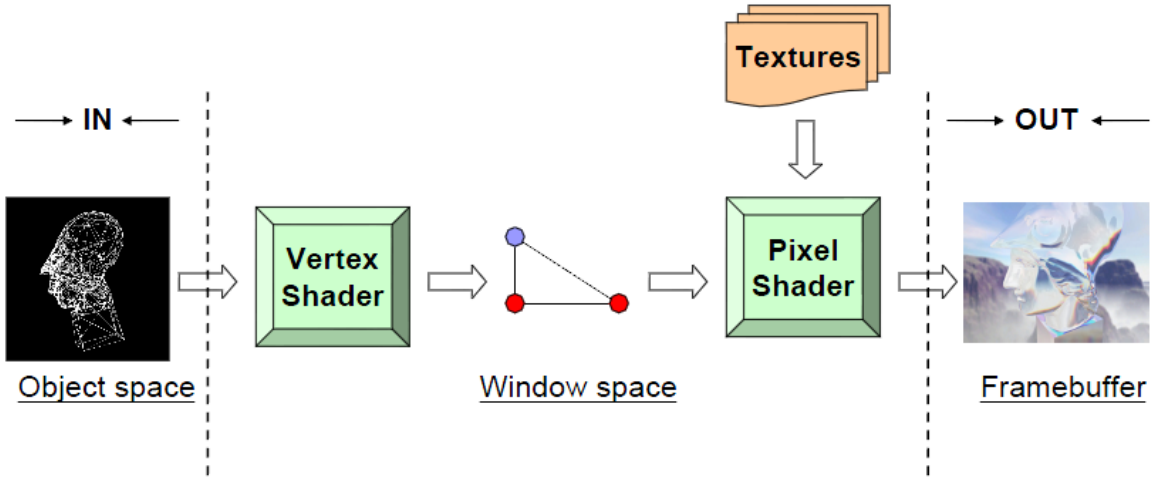


GPGPU: primeros programas de propósito general

- Las **GPUs**:
  - son altamente paralelas
  - tienen gran núm. *shaders* (16-200)
  - pueden ejecutar muchos **hilos**
  - comparativamente muy **baratas**
- Idea:
  - *Shader* como motor de computación no gráfico
  - Primer enfoque: convertir datos a formato de **gráficos** y aplicar transformaciones

Evolución a lenguajes de alto nivel

- Surge la necesidad de pasar a lenguajes de **alto nivel**
- Facilitar la programación de **shaders**
- Surgen los lenguajes Cg, HLSL, GLSlang, CTM…



NVIDIA CUDA

- CUDA = **C**ompute **U**nified **D**evice **A**rquitecture
- "*Una plataforma diseñada conjuntamente a nivel software y hardware que permite al programador aprovechar la potencia de una GPU en aplicaciones de propósito general.*"

Introducción a GPU Computing

Introducción a CUDA

Introducción a CUDA

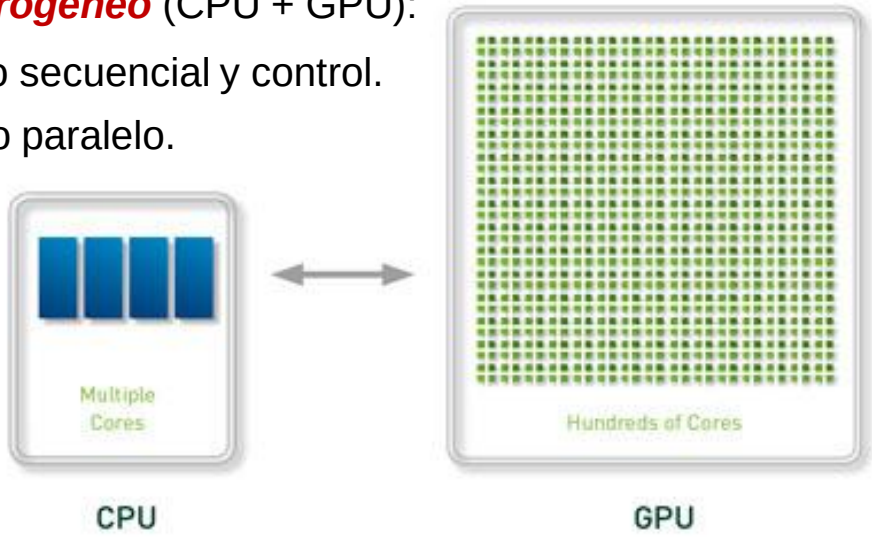
- CUDA hace referencia a:
  - Modelo de programación paralela.
  - Entorno de programación: extensión de C/C++.
  - Compilador, driver y conjunto de herramientas.
  - Dispositivos y arquitecturas GPU compatibles.

Introducción a CUDA

- Ofrece:
  - **Computación paralela para las masas**: solución bajo coste/rendimiento (280€/3.2TFLOPS, GeForce GTX770).
  - Computación masivamente paralela: nuevo modelo de propósito general de “alto nivel”.

Introducción a CUDA

- CUDA permite:
  - **Escalar** el paralelismo a miles de hilos “ligeros” ejecutándose en cientos de procesadores.
  - **Abstraer** la GPU a los programadores.
  - Componer un ***sistema heterogéneo*** (CPU + GPU):
    - CPU dedicada a código secuencial y control.
    - GPU dedicada a código paralelo.

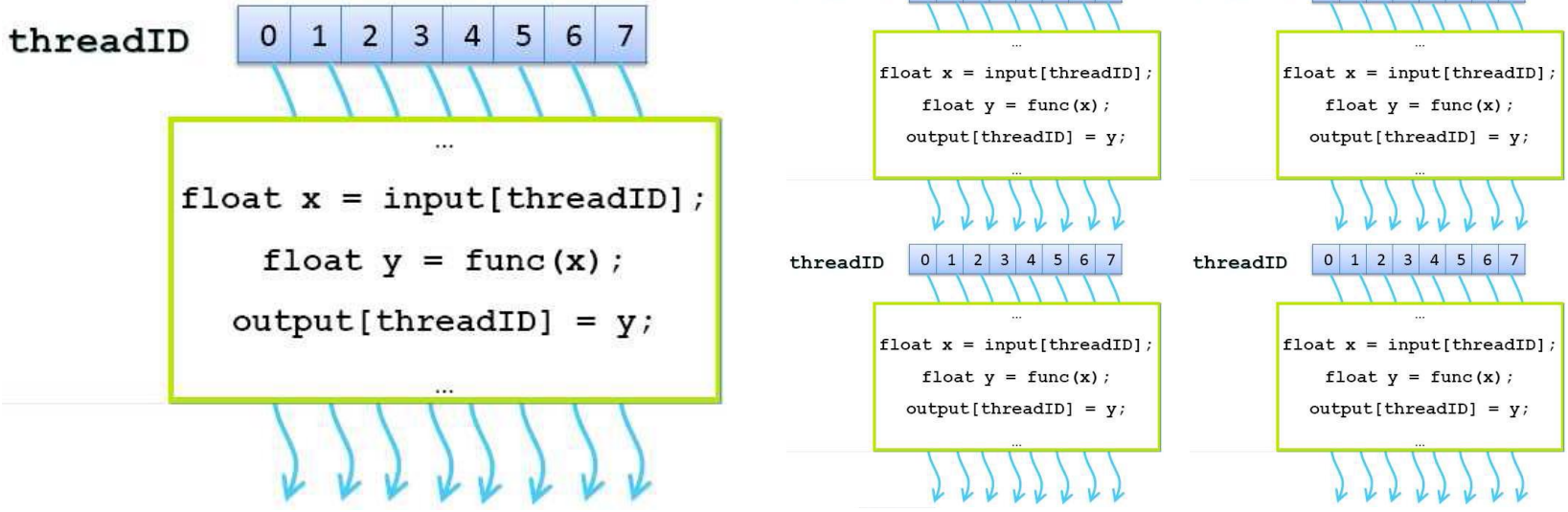


Introducción a CUDA

- Algunas **definiciones** del modelo:
  - *Host, Device, Kernel*
  - *Compute Capability*
  - *Warp (32 hilos)*
- Los hilos son muy **ligeros**: poca sobrecarga de creación y de planificación.
- Primitivas de **sincronización** simples y ligeras.

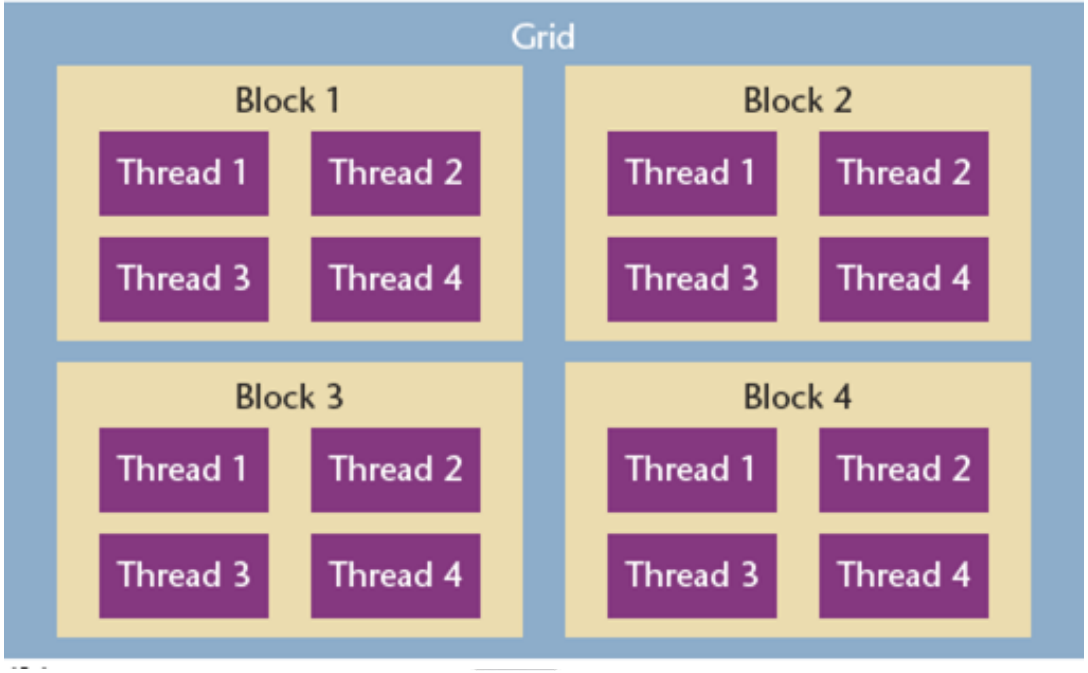
Introducción a CUDA

- La ejecución de los hilos es **SPMD** (**S**ingle **P**rogram **M**ultiple **D**ata): ejecutan el código (del mismo kernel) sobre distintos datos (copiados previamente a la GPU).



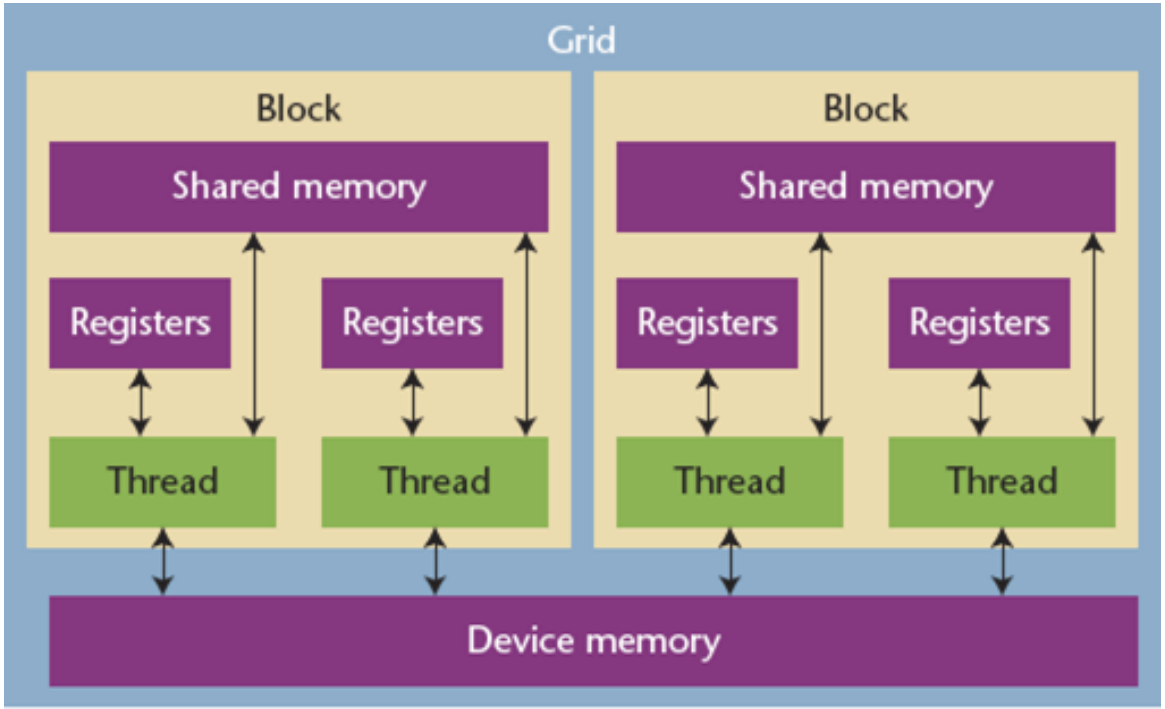
Introducción a CUDA

- **Jerarquía** de **hilos** en el modelo.



Introducción a CUDA

- **Jerarquía** de **memoria** en el modelo:



Tarjetas gráficas de NVIDIA

- La arquitectura “Kepler”, K40 (2013):
  - **15** SMX x **192** SP, 745Mhz
  - **65536** registers.
  - **48** KB shared/L1.
  - **1.5** MB L2 cache.
  - **12** GB global ECC.

Tarjetas gráficas de NVIDIA

- **Gamas** de NVIDIA:
  - **GeForce**: gráficos de consumo (videojuegos).
  - **Quadro**: visualización profesional.
  - **Tesla**: cálculo paralelo y computación de altas prestaciones.
  - **Ion**: para portátiles.
- En común: Todas las soluciones **soportan** CUDA.
- Tesla es dedicada (más recursos), tiene ECC y soporte de NVIDIA para centros de computación.

Desarrollo en CUDA

- Lenguaje CUDA C++:
  - Corta línea de aprendizaje (una librería sobre C++).
  - Permite al programador involucrarse a distintos grados de exigencia, según el nivel de rendimiento deseado:
    - **Básico**: Fácil portabilidad desde C/C++.
    - **Medio**: Requiere un buen conocimiento de la arquitectura gráfica subyacente.
    - **Avanzado (ninja)**: Mapeo eficiente del problema sobre muchos procesadores SIMD, minimizando conflictos en el acceso a memoria.

Desarrollo en CUDA

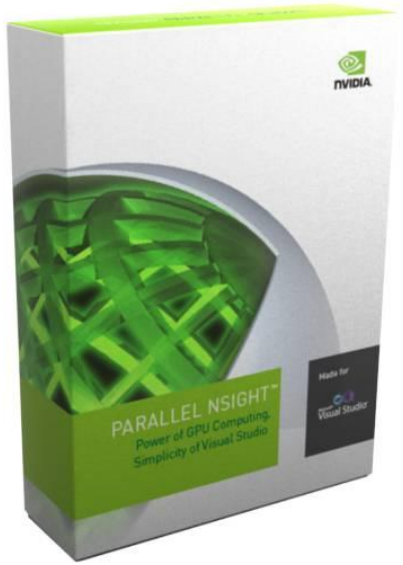
- El software de desarrollo en 3 **categorías**:
  - **"Baja abstracción"**: Programación clásica en CUDA C/C++ (CUDA Toolkit, CUDA SDK y Parallel Nsight).
  - **"Subrutinas estándar de alta abstracción"**: Bibliotecas desarrolladas en CUDA que implementan soluciones a problemas bien conocidos (CUDPP, CUFFT, CUBLAS, CURAND...).
  - **"Aproximación de compiladores de alta abstracción"**: Lenguajes de alto nivel, con compiladores, que evitan la sobrecarga de código de CUDA (PGI, HMPP, PyCUDA...).

Desarrollo en CUDA

- CUDA **Toolkit** 6.5:
  - Compilador nvcc para C/C++, depurador cuda-gdb
  - Visual Profiler, Parallel Nsight
  - Librerías estándar: BLAS, FFT, SPARSE, RAND, MATH, THRUST...
  - Documentación.
- CUDA **SDK** 6.5:
  - Códigos de ejemplo: matrices, vectores, fluidos, etc.
  - Algunas librerías para desarrollo en CUDA.
- *http://developer.nvidia.com/*

Desarrollo en CUDA

- **Parallel Nsight:**
  - **Windows 7/8**: Extensión para MS Visual Studio.
  - **Linux**: Extensión para Eclipse.
  - Compilación y depuración visual.
  - Analizador (versión profesional).
  - (Win) Uso servidor GPU remoto (monitor).

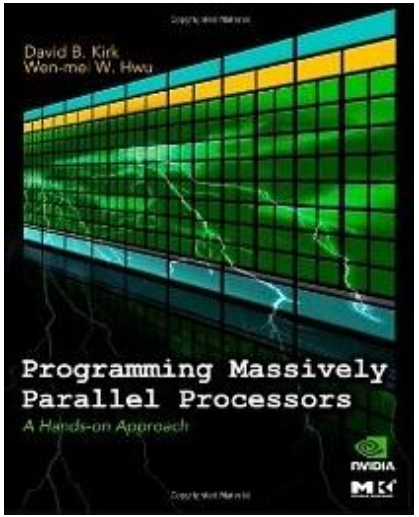


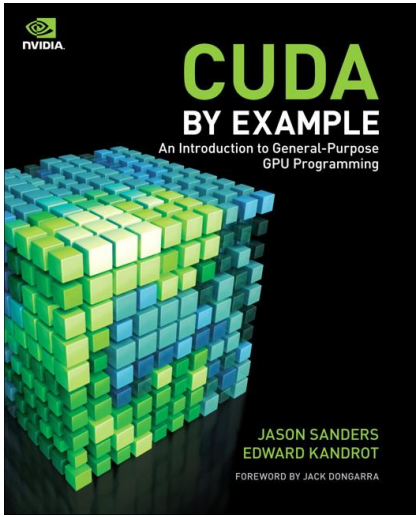
Cómo aprender CUDA

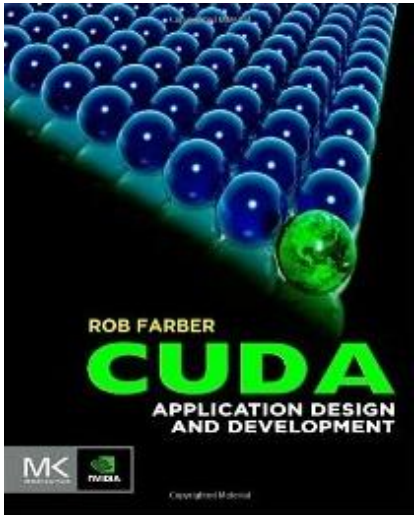
- Página web oficial y programming model guide: ***https://developer.nvidia.com/cuda-zone***
- Entrenamiento y cursos: *https://developer.nvidia.com/cuda-training*
- Universidades: *http://www.nvidia.com/object/cuda_courses_and_map.html*
- Seminarios gratuitos: *http://www.gputechconf.com/resources/gtc-express-webinar-program*

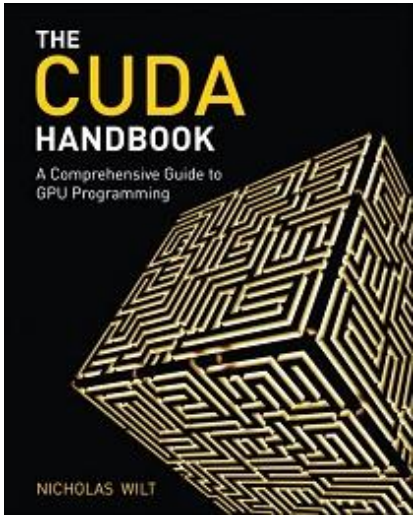
Cómo aprender CUDA

- Libros recomendados:





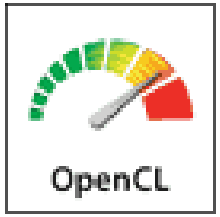




Introducción a GPU Computing

Otros estándares y tecnologías

Introducción a OpenCL

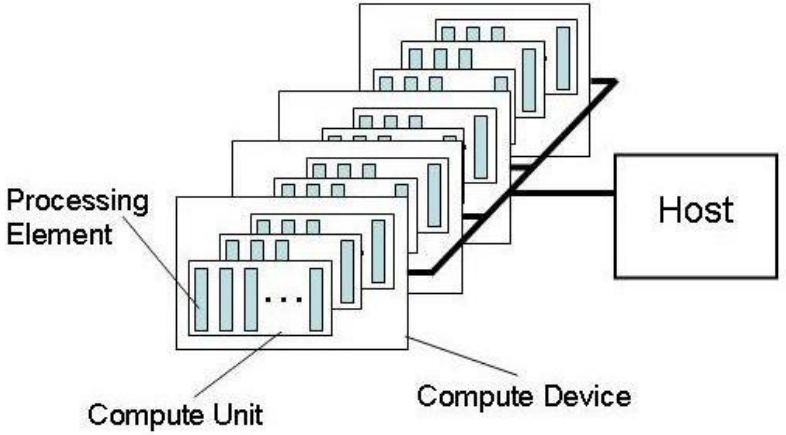


- **Open C**omputing **L**anguage.
- Primer estándar para programación de sistemas heterogéneos basados en GPGPU y C/C++.
- Por el **consorcio** Khronos Group:
  - Más de 100 compañías: Intel, Nvidia, ATI, Apple, ...
  - Creador APIs gráficas y cómputo paralelo: OpenGL, OpenAL, ...
- Introducido por Apple. Aprobado y adaptado por Khronos.

KHRONOS GROUP

Introducción a OpenCL

- Device: GPU y/o CPU.
- Código no compilado (en runtime).
- **Similitudes** con CUDA:
  - Hilo = Item de trabajo
  - Bloque de hilos = Grupo de trabajo
  - Kernel = Kernel
  - Shared memory = Local memory
  - Global memory = Global memory



Programación en OpenCL

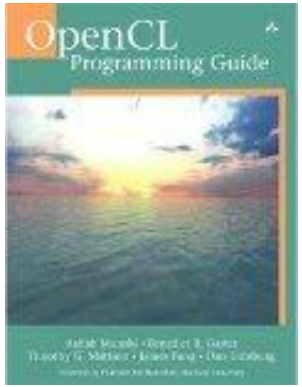
- Soporte de **NVIDIA**:
  - Compilador nvcc adaptado.
  - Librerías del toolkit y códigos del SDK ampliados con OpenCL.
  - Código ejecutado sobre CUDA: pérdida rendimiento.
- Soporte de **AMD**:
  - Stream SDK para gamas FirePro y Radeon (2816 cores con 16 GB de memoria).
  - AMD APP (Accelerated Parallel Processing).
- Soporte de **Intel**:
  - Compilador para extensiones AVX de Intel Core e Intel Xeon.
  - Haswell, Ivy Bridge, Intel Phi

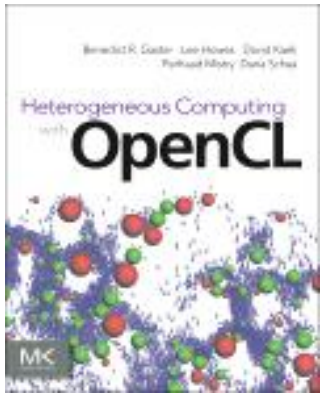
Cómo aprender OpenCL

- Página de desarrollo OpenCL de NVIDIA:
  https://developer.nvidia.com/opencl
- Página desarrollo AMD:
  http://developer.amd.com
  http://developer.amd.com/tools-and-sdks/opencl-zone/
- Seminarios gratuitos:
  https://www.youtube.com/playlist?list=PLVk9nlso0x0LoT7P-GVWEASBSS750gLn6

Cómo aprender OpenCL

- Libros **recomendados**:





Programación en OpenAcc

- Tecnología de Cray, CAPS, NVIDIA y PGI
- Basado en directivas (estilo OpenMP).
- Soporte sólo comercial (NVIDIA PGI), aunque GNU GCC trabaja en incluirlo.

```
#pragma acc kernels loop gang(32), vector(16)
        for( int j = 1; j < n-1; j++)
        {
#pragma acc loop gang(16), vector(32)
            for( int i = 1; i < m-1; i++ )
            {
                Anew[j][i] = 0.25f * ( A[j][i+1]
                                     + A[j-1][i]
                error = fmaxf( error, fabsf(Anew
            }
        }
```

OpenACC
Directives for Accelerators

NVIDIA CUDA RESEARCH CENTER

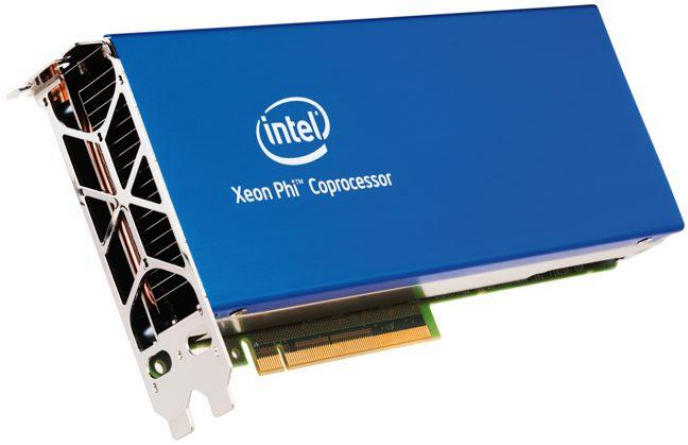
RGNC

AMD FirePro, Radeon y Fusion

- AMD adquirió ATI, mayor competidor en GPUs
- FirePro S9150: 2815 núcleos, 16GB memoria, 5TFLOPS
- Programado con OpenCL y Microsoft C++ AMP
- Algunas “ventajas”:
  - Suelen ser más baratas
  - Mejor rendimiento de cálculo
  - APUs avanzadas

Intel Xeon Phi (MIC)

- Intel **M**any **I**ntegrated **C**ore **A**rchitecture (Intel MIC) es un multiprocesador empaquetado en una tarjeta, que sirve como coprocesador.
- En procesadores Intel Xeon Phi
- 61 cores, 4 hilos por core: 244 hilos
- Programación con TBB, OpenCL



Procesadores híbridos

- Procesadores de última generación que **combinan** CPU y GPU en el mismo chip.
  - Controlador implementado en chip.
  - Comparte misma memoria (RAM, más lenta: DDR3 vs GDDR5).
- Algunas tecnologías:
  - NVIDIA Tegra (móviles y tablets): 4CPUs (ARM) + 1GPU (192 cores) (Tegra K1)
  - AMD Fusion (APU): 4CPUs + 8GPU (512 cores) (A10-7850K)
  - Intel Haswell (22nm): 4(8) + 1GPU (20 cores)

gpgpu.org

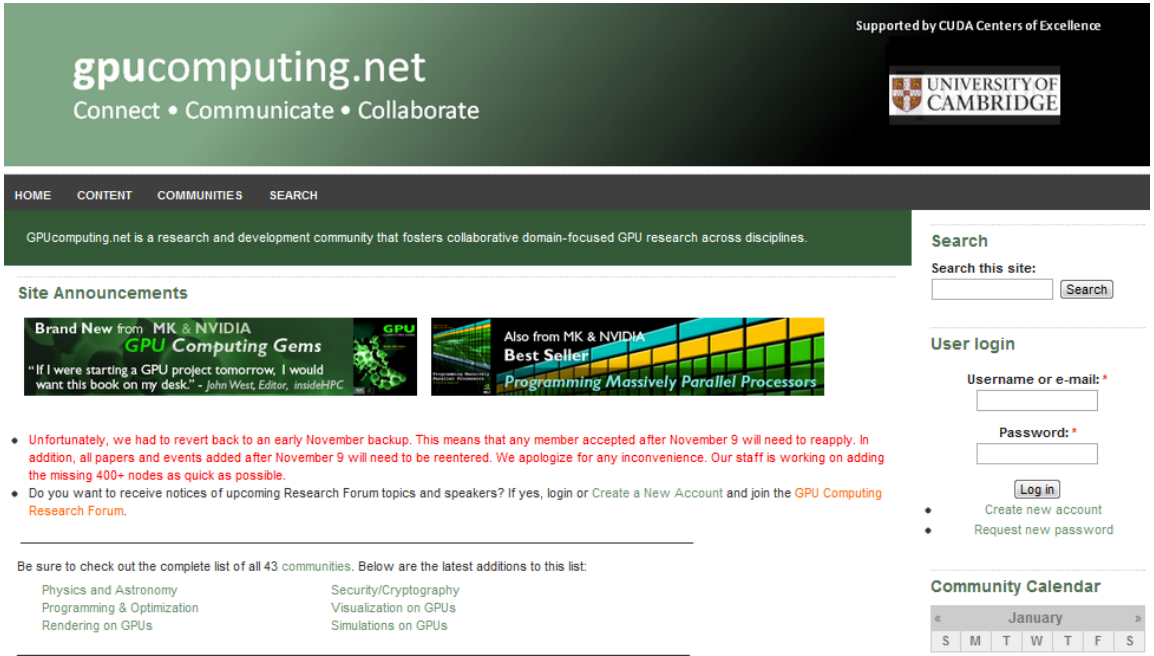
- Recursos para desarrolladores, noticias, eventos, foros, calls…

hgpu.org

- Base de datos que contiene trabajos relacionados con GPUs

gpucomputing.net

- Más información sobre recursos y comunidades



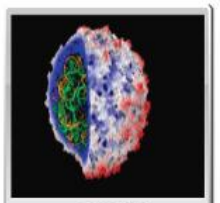
Recursos y oportunidades

Aplicaciones de la GPU

146X

Imágenes biomédicas

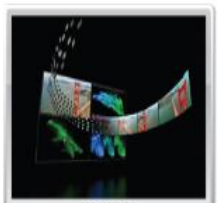
Univ. Utah



36X

Dinámica molecular

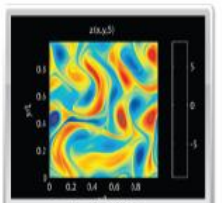
Univ. Illinois, Urbana



18X

Transcoding de video

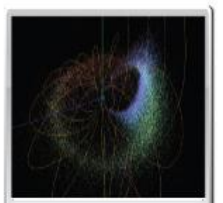
Elemental Tech



50X

Computación Matlab

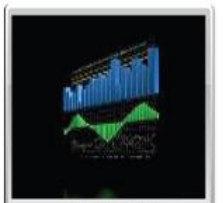
AccelerEyes



100X

Astrofísica

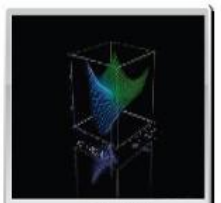
RIKEN



149X

Simulación financiera

Oxford



47X

Algebra lineal

Univ. Jaume I

20X

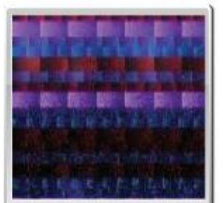
Ultrasonidos 3D

Techniscan

130X

Química cuántica

Univ. Illinois, Urbana



30X

Secuenciación genética

Univ. Maryland

Supercomputación y CUDA

- Top500, Junio 2014: 2º y 6º puestos.

RANK	SITE	SYSTEM	CORES	RMAX (TFLOP/S)	RPEAK (TFLOP/S)	POWER (KW)
1	National Super Computer Center in Guangzhou China	**Tianhe-2 (MilkyWay-2)** - TH-IVB-FEP Cluster, Intel Xeon E5-2692 12C 2.200GHz, TH Express-2, Intel Xeon Phi 31S1P NUDT	3120000	33862.7	54902.4	17808
2	DOE/SC/Oak Ridge National Laboratory United States	**Titan** - Cray XK7 , Opteron 6274 16C 2.200GHz, Cray Gemini interconnect, NVIDIA K20x Cray Inc.	560640	17590.0	27112.5	8209
3	DOE/NNSA/LLNL United States	**Sequoia** - BlueGene/Q, Power BQC 16C 1.60 GHz, Custom IBM	1572864	17173.2	20132.7	7890
4	RIKEN Advanced Institute for Computational Science (AICS) Japan	K computer, SPARC64 VIIIfx 2.0GHz, Tofu interconnect Fujitsu	705024	10510.0	11280.4	12660
5	DOE/SC/Argonne National Laboratory United States	**Mira** - BlueGene/Q, Power BQC 16C 1.60GHz, Custom IBM	786432	8586.6	10066.3	3945
6	Swiss National Supercomputing Centre (CSCS) Switzerland	**Piz Daint** - Cray XC30, Xeon E5-2670 8C 2.600GHz, Aries interconnect , NVIDIA K20x Cray Inc.	115984	6271.0	7788.9	2325

Recursos en el RGNC

- Servidor “Teide”:
  - 3 x Tesla C1060 (240 cores @ 1.3Ghz, 4GB)
  - 1 x GeForce GTX 550Ti (192 cores @ 1.9Ghz, 1GB)
- Servidor “Mulhacen”:
  - 1 x Tesla K40 (2880 cores @ 0.75Ghz, 12GB)
  - 1 x GeForce GTX780 Ti (2880 cores @ 0.93Ghz, 3GB)
  - 1 x GeForce 9400GT (16 cores @ 1.4Ghz, 512MB)
- Financiados por MINECO, MICINN y Junta Andalucía
- http://www.gcn.us.es/gpucomputing

Otros recursos

- NVIDIA Test Drive: http://www.nvidia.com/object/gpu-test-drive.html
  - Acceso de prueba a un servidor con GPU K20x, gratuito.
- Amazon EC2: http://aws.amazon.com/es/ec2/
  - Cuenta gratuita, pago de recursos por hora (~0,65$/hora)
- BSC: http://www.bsc.es/marenostrum-support-services
  - Alta demanda y uso restringido.
- CETA-CIEMAT: http://www.ceta-ciemat.es
  - Cluster con GPUs, financiado con fondos públicos, fácil acceso.

Oportunidades

- Con CUDA:
  - Aceleración de aplicaciones a “bajo coste” y en local.
  - Aumento potencia de nodos de computación en clusters.
  - Requiere esfuerzo inicial para adaptar el problema.
- Oportunidades de publicación:
  - En el área correspondiente.
  - En el área de GPU computing.
- *Estamos abiertos a cualquier tipo de colaboración.*

Sería fantástico, dentro de la US, …

- Crear una comunidad de interesados en la tecnología GPU
- Crear una comunidad de desarrolladores en CUDA
- Aunar esfuerzos para formar a los alumnos en nuevas tendencias de supercomputación.

Muchas gracias

- ¿Preguntas?
- Correo electrónico: mdelamor@us.es
- Página web: www.cs.us.es/~mdelamor