Paralelismo

Flores, Facundo Gabriel

Paradigmas y Lenguajes

16-11-2014

Introducción

- Una experiencia humana común.
- La serialización es el acto de poner un conjunto de operaciones en algún orden.

Ejemplo de paralelismo

```
for(i = 0; i < num_web_sites; ++i)
    buscar(frase, website[i]);
```

Ejemplo de paralelismo

```
parallel_for(i = 0; i < num_web_sites; ++i)
    buscar(frase, website[i]);
```

Motivaciones

Ley de Moore

Establece que cada dos años se duplica el número de transistores en un circuito integrado.

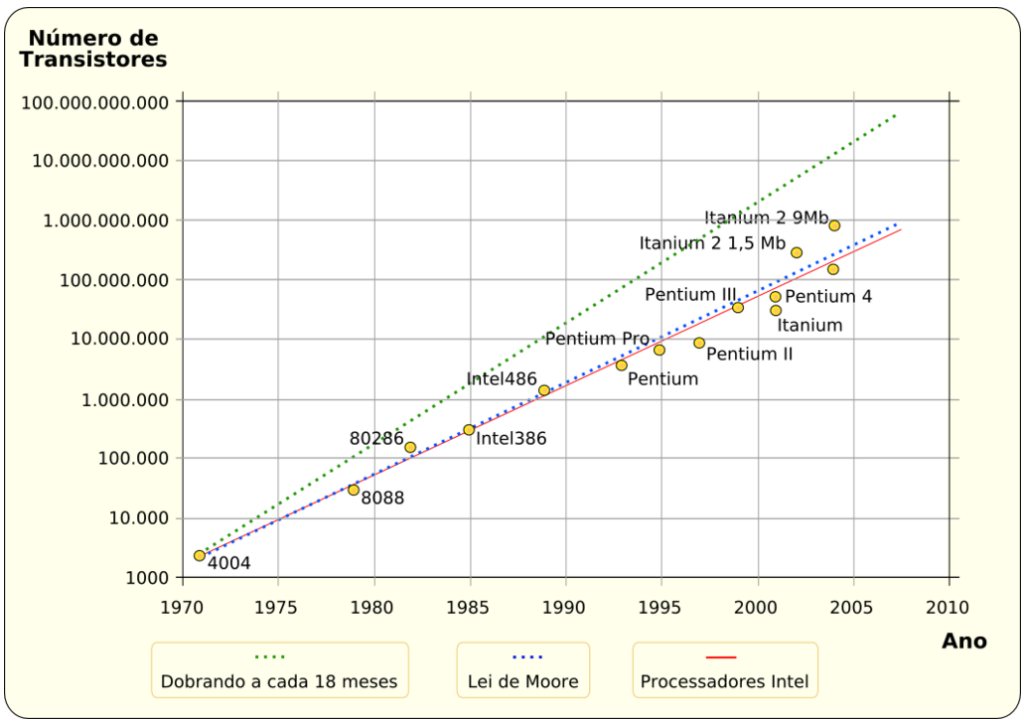


Figura 1 : Número de procesadores a través del tiempo.

Motivaciones

Sin embargo...

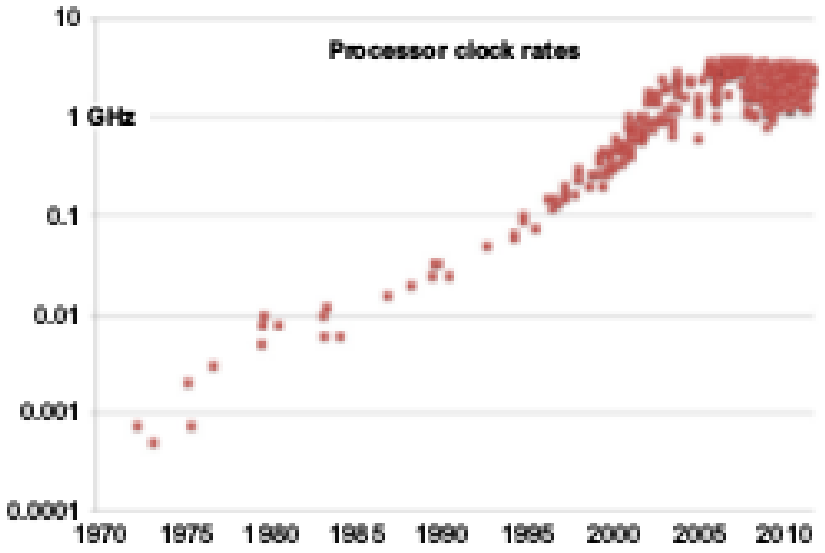


Figura 2 : Crecimiento de las velocidades de procesador en el tiempo.

Qué sucedió?

- **Power wall:** El crecimiento en el uso de energía por la velocidad del reloj.
- **Instruction-lvel parallelism (ILP) wall:** Limitada disponibilidad de paralelismo a bajo nivel.
- **Memory wall:** Discrepancia entre el crecimiento de velocidad de procesadores con respecto a las velocidades de memoria.

Modelos de programación

- **TBB** Threading Building Blocks, soporta modelos de paralelismo basados en el modelo de tareas. Cuenta con las siguientes características:
  - Soporte directo para patrones paralelos como ser: map, fork-join, task graphs, reducción, scan.
  - Una buen manejo de balanceo de carga
  - Una colección de estructuras de datos thread-safe.

Modelos de programación

- **OpenMP**Open Multi-Processing. Cuenta con las siguientes características:
  - Creación de equipos de threads que ejecutan un bloque de código en conjunto.
  - Conversión de "loopsçon extensiones acotadas a ejecución paralela por un equipo de threads con una sintaxis bastante simple.
  - Soporte para la ejecución de un modelo de tareas por la explicitación de un equipo de threads.

Modelos de programación

- **OpenCL**Open Compute Language. Cuenta con las siguientes características:
  - Habilidad de descargar computaciones y datos a un co-procesador con unidad de memoria separada.
  - Invocación de una grilla para operaciones paralelas usando funciones "kernel".
  - Soporte para una cola de tareas que pueden manejar invocaciones de kernels en forma paralela.

¿Cuándo utilizar cada modelo?

- TBB puede ser utilizado cada vez que sea necesaria una solución portable. Un buen modelo para C++.
- OpenMP está disponible universalmente para compiladores Fortran, C, y C++. Predonamina sobre "loops" intensamente computacionales.
- OpenCL provee una solución estandarizada para la interacción entre GPUs, CPUs y aceleradores.

Speedup, Eficiencia, Escalabilidad

Speedup: Sea T_1 el tiempo de ejecución de la aplicación con un procesador y T_P el tiempo de ejecución de una aplicación con P procesadores:

$$\text{speedup} = S_P = \frac{T_1}{T_P}$$

Speedup, Eficiencia, Escalabilidad

Eficiencia: es el speedup dividido el número de procesadores

$$\text{eficiencia} = \frac{S_P}{P} = \frac{T_1}{PT_P}$$

Speedup, Eficiencia, Escalabilidad

Escalabilidad: Si el programa aumenta su speedup cuando P tenemos una **escalabilidad fuerte**. Por otro lado, si el tamaño del problema crece tanto como crece P, sin aumentar la escalabilidad, estamos frente a una **escalabilidad débil**.

¿Qué es?

El cálculo acelerado en la GPU puede definirse como el uso de una unidad de procesamiento gráfico (GPU) en combinación con una CPU para acelerar aplicaciones de empresa, consumo, ingeniería, análisis y cálculo científico. NVIDIA lo introdujo en 2007 y, desde entonces, las GPU aceleradoras han pasado a instalarse en centros de datos energéticamente eficientes de laboratorios gubernamentales, universidades, grandes compañías y PYMEs de todo el mundo.

¿Qué es?

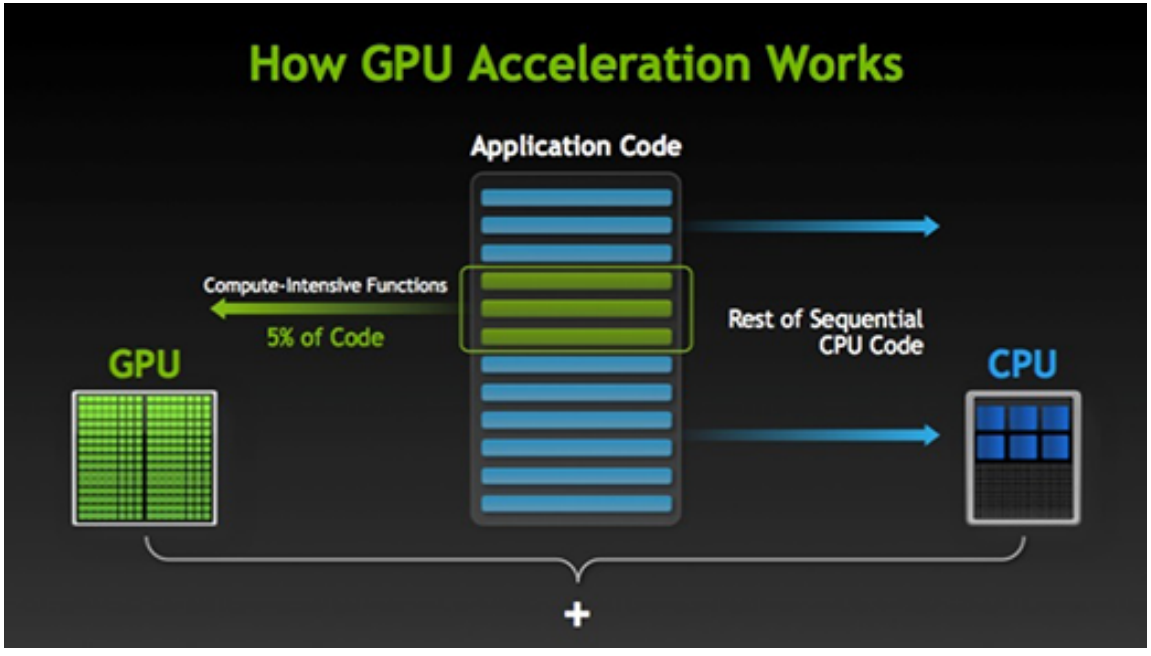


Figura 3 : Cómo las GPUs aceleran el trabajo

¿Qué es?

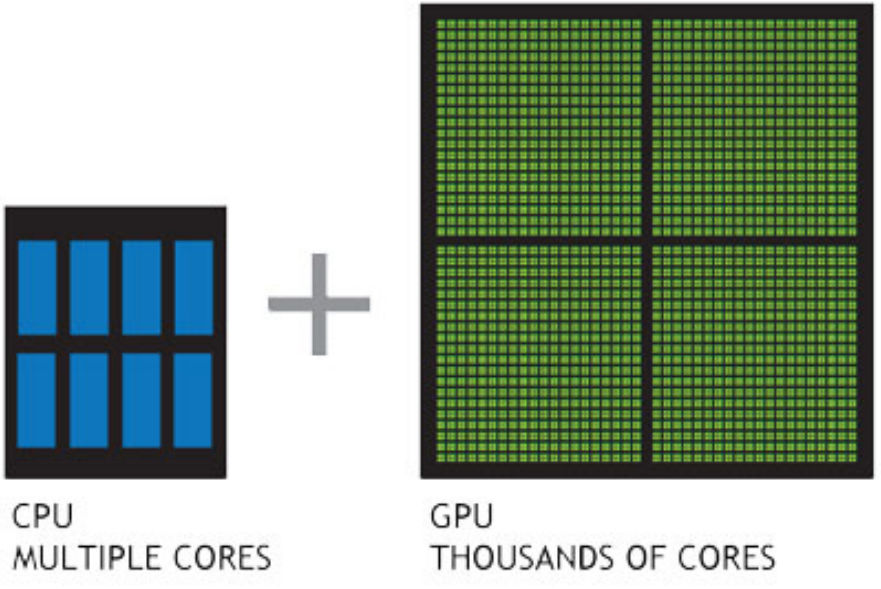
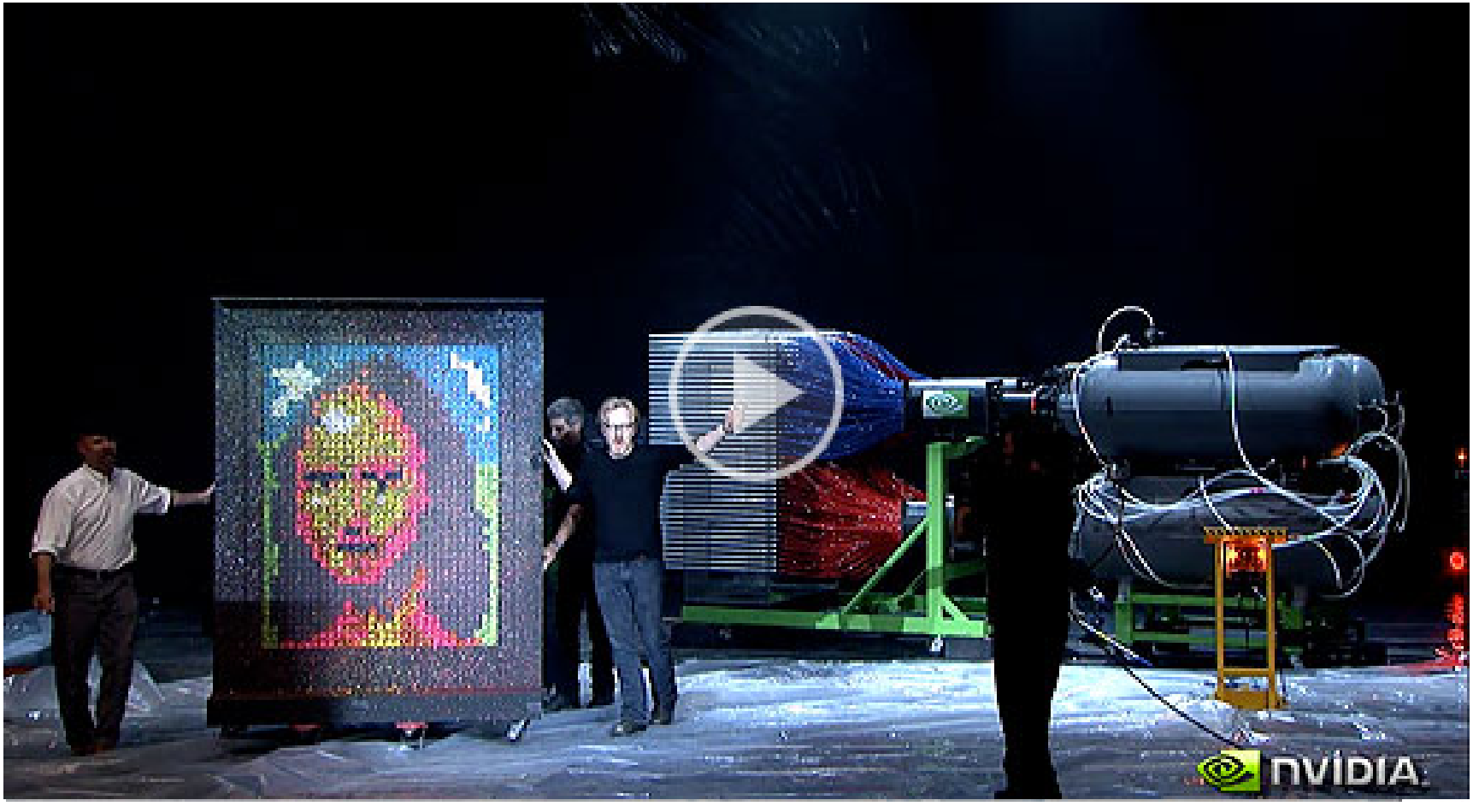


Figura 4 : Comparación entre una CPU y una GPU

Demostración



CUDA

Compute Unified Device Architecture Hace referencia tanto a un compilador como a un conjunto de herramientas de desarrollo creadas por nVidia que permiten a los programadores usar una variación del lenguaje de programación C para codificar algoritmos en GPU de nVidia.