UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
Facultad de Ingeniería

Introducción a Cómputo Paralelo con CUDA C/C++

LABORATORIO DE INTEL Y CÓMPUTO DE ALTO DESEMPEÑO

Elaboran: Ariel Ulloa Trejo
Jaime Beltrán Rosales
Revisión: Ing. Laura Sandoval Montaño

Temario

1. Antecedentes
   - Programación Serial
   - Programación Paralela
   - Modelos de Programación
2. El GPU
   - Arquitectura
   - Diferencias entre el GPU y CPU
   - Propiedades del GPU
3. Modelo de programación CUDA
   - Jerarquia de memoria
     - Mallas
     - Bloques
     - Hilos
   - Funciones CUDA
   - Arreglos unidimencionales
4. Manejo de Matrices
   - Detectando errores
   - Tomando el tiempo
5. Memoria Compartida
6. Varios

1 Antecedentes

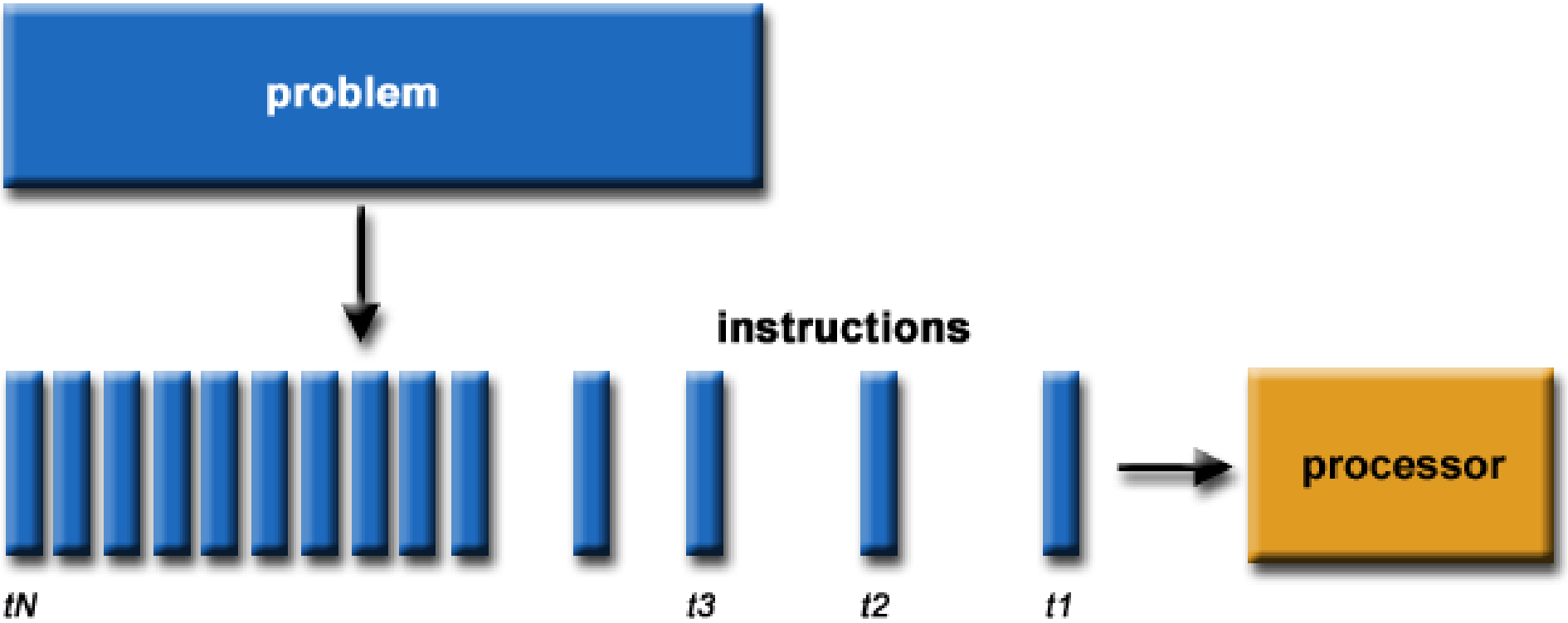
Programación Serial

Desde la invención de la arquitectura Von-Neumann, se optó por el aumento de la frecuencia de los procesadores y en general de los componentes de la computadora para incrementar el rendimiento.
Esto produce que un mayor número de instrucciones sean procesadas en un determinado tiempo.

Pero a su vez crea otros inconvenientes, como el aumento de temperatura y deterioro acelerado de los componentes de los sistemas. Debido a esto se buscaron otras formas de mejorar el rendimiento sin llegar a aumentar las frecuencias a valores exorbitantes.

```c
#include <stdio.h>

main(){
    int i=0;
    if(i)
        printf("1");
    else
        for( ; i < 10000; i++)
            printf("%d\t", i);
    return 0;
}
```

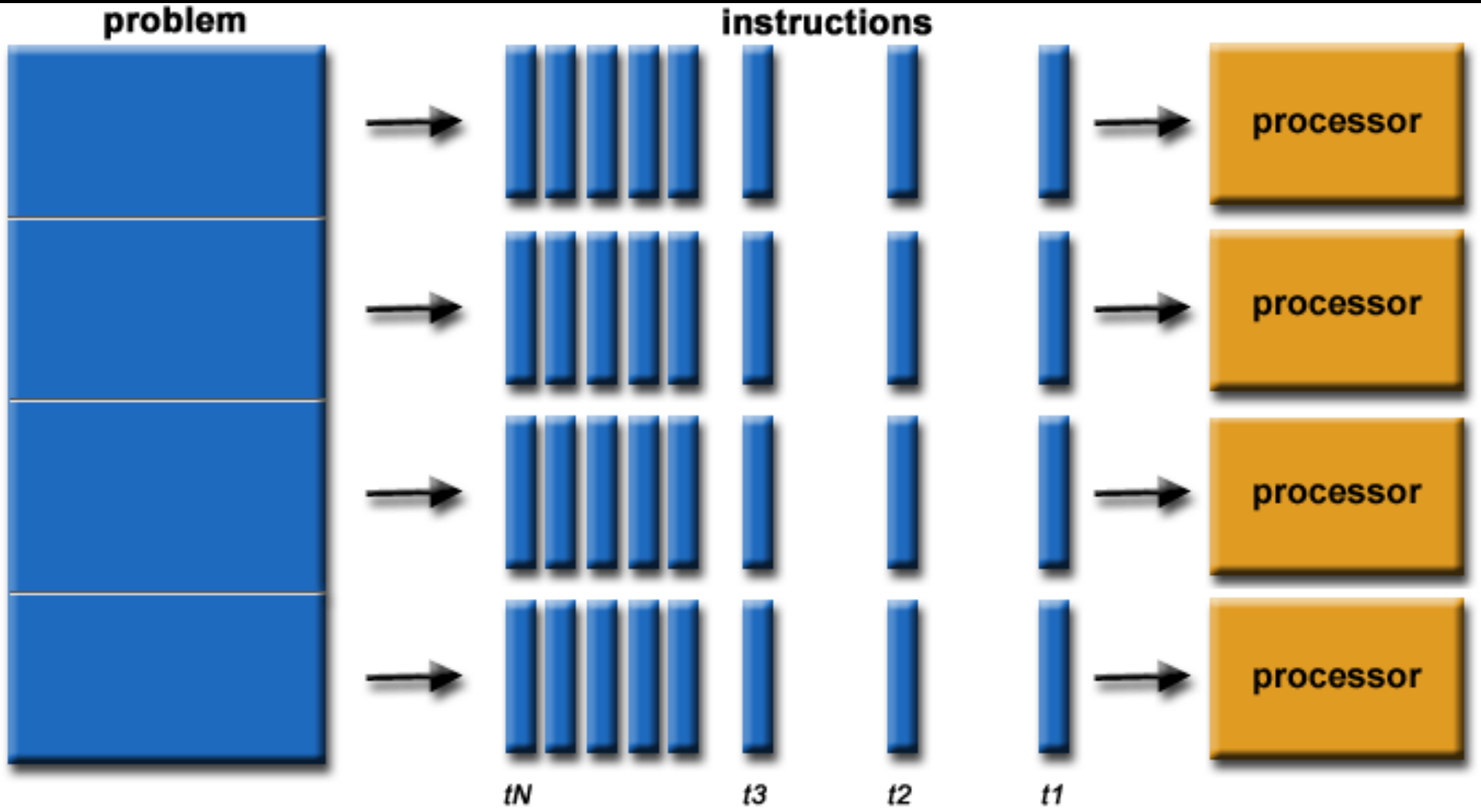

problem
instructions
processor
tN
t3
t2
t1

1 Antecedentes

Programación Paralela

Es el uso simultáneo de múltiples recursos de la computadora para resolver un problema computable:

- Un problema es separado en partes que pueden resolverse concurrentemente
- Cada parte es dividida en una serie de instrucciones.
- Las instrucciones de cada parte se ejecutan simultáneamente en diferentes procesadores.
- Se emplea un mecanismo de control/coordinación.


problem
instructions
processor
processor
processor
processor
tN
t3
t2
t1

1 Antecedentes

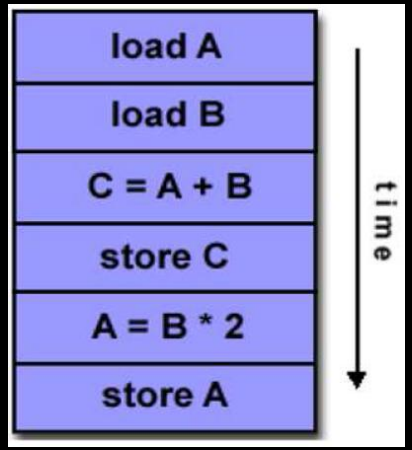
Modelos de Programación

La taxonomía de Flynn es una clasificación de las diferentes arquitecturas que implementan los procesadores basada en la forma en que procesan los datos y las instrucciones.

1 Antecedentes

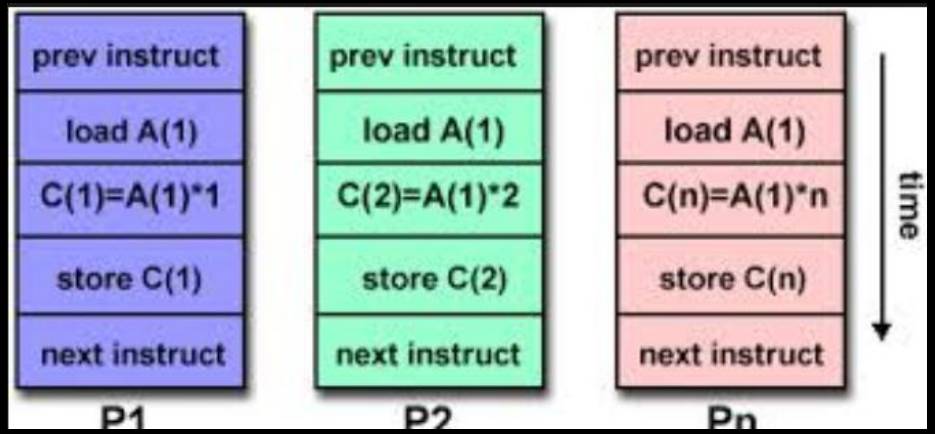
Single Instruction, Single Data (SISD)

Es el modelo de una computadora serial (un sólo CPU).



Multiple Instruction, Single Data (MISD)

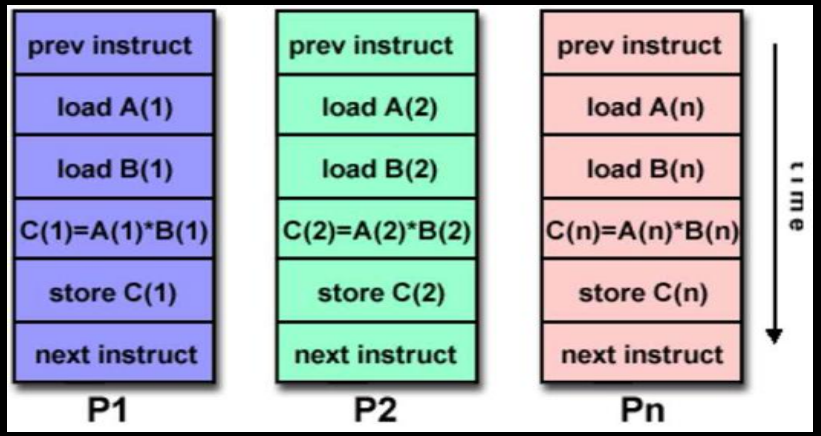
Un conjunto de datos, utilizando instrucciones diferentes simultáneamente.



1 Antecedentes

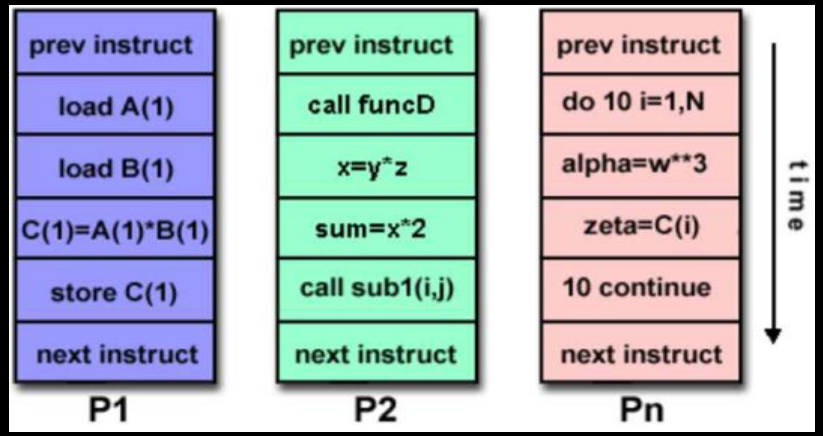
Single Instruction, Multiple Data (SIMD)

Conjuntos de datos diferentes son procesados simultáneamente por las mismas instrucciones.



Multiple Instruction, Multiple Data (MISD)

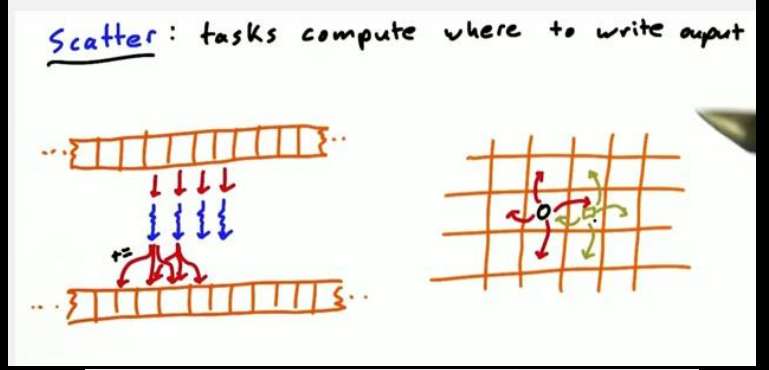
Conjuntos de datos diferentes son procesados simultáneamente por diferentes instrucciones.

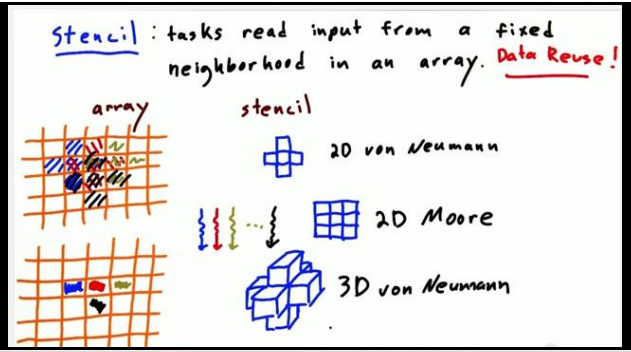


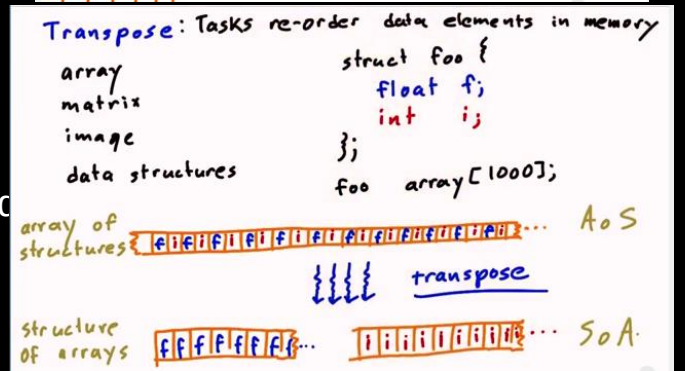
Antecedentes

Conceptos

- Dispositivo (device)
- CPU (host)
- GPU
- Hilos (Threads)
- Bloque (block)
- Sincronizar hilos (Synchronize threads)
- Kernel
- Ancho de banda (Rendimiento)
- Latencia (Latency)
- Mapeo (map) (uno a uno)
- Reunir (gather) (muchos a uno)
- Dispersar (scatter) (uno a muchos)
- Plantilla (Stencil))(varios a uno, normalmente usando ac
- Memoria GPU (local, compartida y global)







1 Antecedentes

¿Por qué programación paralela?

Disminuir tiempo y optimizar recursos:

- En teoría, más recursos para una tarea disminuyen su tiempo de finalización, con un ahorro potencial de costos

Resolver problemas más grandes y complejos:

- Muchos problemas son tan grandes o complejos que es impráctico o casi imposible resolverlos en un solo procesador, especialmente con memoria limitada, como por ejemplo los motores de búsquedas web o de bases de datos, que necesitan procesar millones de transacciones por segundo.

Construir computadoras más poderosas:

- Anteriormente construimos mejores computadoras aumentando la velocidad del reloj, sin embargo actualmente esto ya no es posible debido a que nos encontramos cerca de la barrera de la potencia