



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE CIENCIAS

Misión Secundaria

EQUIPO

**Arroyo Martínez Erick Daniel
Terrazas Rivera Alex
Vergara Navarro Mixtli Arturo**

PROFESOR

Gilde Valeria Rodríguez Jiménez

AYUDANTES

**Rogelio Alcantar Arenas
Gibran Aguilar Zuñiga
Luis Angel Leyva Castillo
Rogelio Alcantar Arenas**

ASIGNATURA

Computación Concurrente

25 de febrero de 2024

Introducción

Este trabajo consiste en la elaboración de un sistema capaz de resolver el producto de matrices, de forma secuencial y paralela. Con el propósito de analizar el rendimiento del sistema paralelo con respecto al secuencial.

Especificaciones

Como se pudo observar en el código, la carga de valores pseudoaleatorios a cada entrada de las matrices A y B se realiza de forma paralela, pues le asignamos a cada hilo una fila de cada matriz. Además, asumimos que la prueba con un solo hilo corresponde al algoritmo secuencial, también realizado. Para las siguientes dos pruebas 100 y 1000 hilos, la métrica que usamos para evaluar estos casos es, dada nuestra implementación, necesitamos $A_{filas} * B_{columnas}$ hilos, esto por conveniencia. Por lo tanto, para los últimos dos tests, bastarían matrices de 10x10 y 100x100, sin obviar que el sistema puede trabajar con cualesquiera dimensiones, siempre y cuando respeten la definición del producto.

Análisis

Dado que la creación aleatoria de las entradas de las matrices es independiente del cálculo del producto. Entonces, para determinar el porcentaje paralelizable de nuestro código usamos la fórmula:

$$T_{paralelo} = \frac{T_{Total} - T_{calculo}}{T_{Total}} \quad (1)$$

Estos datos los obtuvimos por medio de determinar el tiempo que tarda un solo hilo en calcular una entrada (esto como media) con respecto a los N hilos requeridos que determinan todas las entradas. Entonces, sustituyendo tenemos:

$$T_{paralelo} = \frac{0,6023900000000001 - 1,5100000000000001 \times 10^{-5}}{0,6023900000000001} \approx 0,999975 \quad (2)$$

Esto quiere decir que, prácticamente la completitud de nuestro código es paralelizable. Entonces, ya podemos determinar las entradas de nuestra tabla:

# Hilos	Aceleración Teórica	Aceleración Obtenida	% Código Paralelo
1	1	1	99
100	50.2545	30.5555	99
1000	90.9918		183.3333 99

Cuadro 1: Análisis

Conocimiento Adquiridos

Con la realización de este trabajo hemos podido observar uno de los contextos con mayor oportunidad para paralelizar un sistema. Pues, al delegarle a cada proceso una tarea independiente a la del resto, sin obviar nuestra capacidad de cómputo y recursos, podemos elevar el rendimiento de un sistema. Lo ideal es identificar las secciones que permiten esta intrusión de comportamiento paralelo. Como era de esperarse, a los códigos secuenciales (iterativos) deben realizarse aún más operaciones que un sistema paralelo. En este caso, dado que la cantidad de hilos depende del ejemplar dado, podemos observar, de alguna manera, como la cantidad de ellos es directamente proporcional a la aceleración teórica. Pero no podemos garantizar esto para todos los escenarios, pues ya mencionamos que la cantidad de recursos disponibles juega un papel principal y muy importante a la hora de analizar y comparar distintas implementaciones.