

PONTIFICIA UNIVERSIDAD CATÓLICA MADRE Y MAESTRA
FACULTAD DE CIENCIAS E INGENIERÍA
ESCUELA DE INGENIERÍA EN COMPUTACIÓN Y TELECOMUNICACIONES

PROGRAMACIÓN PARALELA Y CONCURRENTE

Practica #1

Paralelización de Algoritmos y Análisis de Rendimiento

Objetivo:

- Aplicar la Ley de Amdahl en contextos prácticos.
- Desarrollar programas paralelos usando diferentes modelos de paralelización.
- Evaluar el rendimiento de programas paralelizados y su eficiencia.
- Analizar la escalabilidad de sistemas paralelos.

Instrucciones:

Completa las siguientes actividades teóricas y prácticas. Puedes trabajar en equipo máximo de 2, pero asegúrate de entender cada paso y ser capaz de explicarlo.

Actividades

Parte 1: Ley de Amdahl

Sea un programa con un tiempo de ejecución secuencial de 120 unidades de tiempo, y se sabe que el 85% de su código es paralelizable.

1. Calcula el tiempo de ejecución paralelo **$T_p(n)$** , el speedup **$S(n)$** y la eficiencia **$E(n)$** cuando se ejecuta $n=1,3,6,9,12$ procesadores.
2. Completa la siguiente tabla:

Procesadores n	Tp(n)	S(n)	E(n)
1	100	1.2	1.2
2	57.41	2.09	1.045
4	36.25	3.31	0.8275
8	25.64	4.68	0.585
16	20.33	5.90	0.36875

Parte 2: Programación

Implementación de Algoritmos Paralelos:

Usando Java 21 o superior realiza las siguientes tareas.

Algoritmo a paralelizar: Suma de un arreglo de 1,000,000 números enteros random comprendido entre 1 y 10,000.

1. Genera un archivo con 1,000,000 de registros comprendido entre 1 y 10,000, el cual deberá usar como base para los demás cálculos.
2. Escribe un programa secuencial que sume los elementos de un arreglo de un millón de enteros.
3. Modifica tu programa para que use múltiples hilos o procesos para realizar la suma en paralelo. Divide el arreglo en partes iguales para cada hilo/proceso.
4. Mide y compara el tiempo de ejecución del programa secuencial y del programa paralelo con 2, 4 y 8 hilos/procesos.
5. Completa la siguiente tabla con los tiempos medidos:

Número de Hilos/Procesos	Tiempo de Ejecución Secuencial (s)	Tiempo de Ejecución Paralelo (s)	Speedup	Eficiencia
1 (Secuencial)	5.579 ms	2.942 ms	1.90	1.90
2	5.627 ms	6.675 ms	0.84	0.42
4	6.568 ms	9.267 ms	0.71	0.18
8	5.850 ms	14.713 ms	0.40	0.05

Nota: Si desea puede utilizar Virtual Threads

Entregables:

- Código fuente del programa en Java utilizando. El código debe subirse a un repositorio público en GitHub.
- Documentación en formato README que incluya una breve descripción del programa, instrucciones para compilar y ejecutar el programa, ejemplos de entrada y salida, y análisis del rendimiento en función del número de procesos utilizados. La documentación también debe subirse al repositorio en GitHub.
- El enlace al repositorio público en GitHub debe ser proporcionado en la plataforma de entrega de la tarea. Asegúrate de que el repositorio sea público para poder acceder al proyecto.