

Sistemas Operativos

Lab 2.2

Jorge Alfredo Mayna Flores

Multiplicación de Matrices:

La implementación de la multiplicación de matrices secuencial se realizó sin ningún problema. Para su implementación en paralelo se dividió el peso de las operaciones entre los distintos threads. Esto se logró haciendo que cada thread se encargara de multiplicar todas las filas y columnas que eran múltiplos de la cantidad de threads + id del thread. Si el número de threads era 4 y el id del thread era 1 entonces ese thread multiplica las filas y columnas 1, 5, 9, 13, 17...

El thread con id 2 multiplica 2, 6, 10, 14, 18 . . . y así para todos los threads. De esta manera se dividían equitativamente el trabajo

Luego de implementar la multiplicación de matrices tanto en secuencial como en paralelo se paso a medir los tiempos que tardaban sus respectivas ejecuciones. Para esto se trabajó con una matriz de tamaño 200x200 para ambos casos. Además, se repitió el proceso 20 veces y se obtuvo el promedio del tiempo.

- **Secuencial:**

El tiempo que tardo ejecutar la multiplicación de las matrices fue de 0.044429 segundos.

- **Paralelo:**

Para 2 threads el tiempo fue 0.015694 segundos.

Para 4 threads el tiempo fue 0.008547 segundos.

Para 8 threads el tiempo fue 0.009003 segundos.

Resultados:

Los resultados obtenidos mostraron que el uso de threads redujo el tiempo de ejecución del programa. Esto debido a que divide las tareas y las ejecuta al mismo tiempo en diferentes cores.

El código se corrió en una computadora de 4 cores, puede ser este un motivo para que al usar 8 threads el tiempo no se halla reducido como se esperaba. Sin embargo, el código de gauss.c si mostró mejoras con 8 threads lo cual resulta curioso.