

# Trabajo Práctico 1

## Sistemas Paralelos

### Punto 1

Resolver la siguiente expresión:

$$A.A + A.B + C.D$$

Donde **A**, **B**, **C** y **D** son matrices de **NxN**.

### Solución

#### Secuencial

Dadas las matrices **A**, **B**, **C** y **D** de tamaño **N\*N**, se busca calcular la ecuación **AA + AB + CD**. Lo primero es calcular los valores de las matrices **AA**, **AB**, **CD** en los índices **i**, **j** siendo estas **AA = A[i,j] \* A[i,j]** --- **AB = A[i,j] \* B[i,j]** --- **CD = C[i,j] \* D[i,j]** utilizando una multiplicación de matrices, y antes de aumentar el índice se suman los valores obtenidos anteriormente y se asignan en la matriz **TOTAL** en la misma posición **i**, **j**, o sea, **TOTAL[i,j] = AA + AB + CD**

Cada una de las multiplicaciones se hace en un bloque **for** diferente, para evitar muchos cambios en la memoria al cargar las matrices, sobre todo al crecer el tamaño de las mismas.

En la primera multiplicación, al ser dos veces la misma matriz, se utiliza la función de transponer la matriz, para luego multiplicarla accediendo normalmente en ambos casos.

#### OpenMP

Se engloba el bloque a paralelizar bajo la primitiva *parallel*, se utiliza la primitiva *for* en los bloques iterativos para que se reparta la cantidad de trabajo entre los hilos y por último la primitiva *private* asignándole a cada hilo sus propios índices y auxiliares para realizar las cuentas.

De igual manera que en el programa secuencial, cada una de las multiplicaciones se realiza en un bloque **for** diferente, y realizando la matriz transpuesta de **A**

### Tiempos

Tamaño	Tiempo Secuencial	Tiempo 2 Hilos	Tiempo 4 Hilos
512	1.676513	1,064041	0,554133
1024	28,401217	19,986088	10,427530
2048	246,675260	175,825562	93.019775

#### Speedup

Tamaño	S <sub>p</sub> 2 Hilos	S <sub>p</sub> 4 Hilos
512	1,575609399	3,02547042
1024	1,421049332	2,723676364
2048	1,402954481	2,673359079

Eficiencia

Tamaño	E <sub>p</sub> 2 Hilos	E <sub>p</sub> 4 Hilos
512	0,7878047	0,756367605
1024	0,710524666	0,680919091
2048	0,701477241	0,66833977

## Conclusiones

Al no haber dependencia de datos entre Hilos dado que cada uno calcula una cantidad de posiciones y solo dependen de las matrices A, B, C y D que estas no son modificadas, no difieren tanto los calculos de eficiencia entre 2 y 4 hilos

## Punto 2

Resolver la siguiente expresión:

$$\sum_{i=0}^m \frac{\max M_i - \min M_i}{\text{avg} M_i} \cdot M_i$$

Donde **M<sub>i</sub>** son matrices cuadradas de **NxN**. **minM<sub>i</sub>** y **maxM<sub>i</sub>** son el mínimo y el máximo valor de los elementos de la matriz **M<sub>i</sub>**, respectivamente.

**avgM<sub>i</sub>** es el valor promedio de los elementos de la matriz **M<sub>i</sub>**

## Solución

### Secuencial

Tenemos una arreglo de tamaño de **M** matrices de **N\*N** cada una.

Recorremos este arreglo y en cada matriz buscamos el minimo, el maximo y vamos sumando los valores de cada posicion de la misma para luego dividir ese total por el tamaño de la matriz y obtener un promedio.

Terminado esto realizamos la ecuacion "**(maximo - minimo)/promedio**" la cual una vez calculada, utilizamos su valor para multiplicar la matriz actual. Y por ultimo terminado la multiplicacion avanzamos a la siguiente matriz.

Finalizado el recorrido del vector, lo volvemos a recorrer para obtener la sumatoria de las matrices

## Pthreads

Se divide el vector para que cada thread calcule el máximo, mínimo y promedio de cada matriz, de esta manera no se genera dependencia de datos en ningún momento.

Al terminar de calcular todos los valores, se realiza una barrera para esperar a todos los hilos, y continuar con la multiplicación de esos valores con la matriz. Esto se hace recorriendo por "columnas", siendo que cada hilo va a trabajar sobre una partición de cada matriz. Esto es más eficiente, ya que no es necesario contar con una zona de exclusión mutua.

## Tiempos

Tamaño	Tiempo Secuencial	Tiempo 2 Hilos	Tiempo 4 Hilos
512	0,374618	0,197695	0,123818
1024	1,457437	0,812408	0,550743
2048	5,823712	3,204458	2,168532

## Speedup

Tamaño	S <sub>p</sub> 2 Hilos	S <sub>p</sub> 4 Hilos
512	1,894929057	3,025553635
1024	1,793971798	2,64631053
2048	1,817378165	2,685555021

## Eficiencia

Tamaño	E <sub>p</sub> 2 Hilos	E <sub>p</sub> 4 Hilos
512	0,947464528	0,756388409
1024	0,896985874	0,661577633
2048	0,908689083	0,671388755

## Conclusiones

No hay dependencia de datos dado que cada hilo calcula **M/hilos** matrices, y al tratarse de un arreglo de matrices, se puede trabajar de diferentes maneras, a conveniencia del problema.