

# 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 normal 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**

#### 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.

### Tiempos

Tamaño	Tiempo Secuencial	Tiempo 2 Hilos	Tiempo 4 Hilos
512	1,157355	0,633085	0,327391
1024	9,367766	5,134782	2,773294
2048	75,865119	41,164858	21,877353

#### Speedup

Tamaño	S <sub>p</sub> 2 Hilos	S <sub>p</sub> 4 Hilos
512	1,828119447	3,53508496
1024	1,824374628	3,377848147
2048	1,842958355	3,467746715

## Eficiencia

Tamaño	E <sub>p</sub> 2 Hilos	E <sub>p</sub> 4 Hilos
512	0,914059735	0,88377124
1024	0,912187314	0,844462037
2048	0,921479178	0,866936679

## 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  $M_i$  son matrices cuadradas de  $N \times N$ .  $\min M_i$  y  $\max M_i$  son el mínimo y el máximo valor de los elementos de la matriz  $M_i$ , respectivamente.

$\text{avg} M_i$  es el valor promedio de los elementos de la matriz  $M_i$

## Solución

### Secuencial

Tenemos un arreglo de tamaño de  $M$  matrices de  $N \times N$  cada una.

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

Terminado esto realizamos la ecuación "**(máximo - mínimo)/promedio**" la cual una vez calculada, utilizamos su valor para multiplicar la matriz actual. Y por último terminado la multiplicación 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 la misma cantidad de matrices. Una vez calculado se divide la sumatoria, cada hilo posee un auxiliar local para calcular su sumatoria que luego se suma en una matriz compartida utilizando 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, pero, al haber una exclusion mutua y al ser una muestra tan chica se nota la diferencia de performance entre 2 y 4 hilos, si fuera una muestra mas grande seria como el Ejercicio 1 y seria eficiente tanto para 2 como 4 hilos