**INGENIERÍA DE SOFTWARE**

**Programación en Paralelo**

**Lab3: Programación en paralelo usando OpenMP**

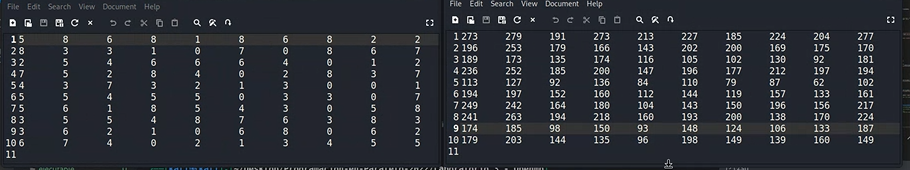
**Juan Pablo Patiño – Natalia Arismendi Pardo**

**Competencias:**

* Reconocer conceptos de programación
* Desarrollar habilidades en programación secuencial y en paralelo usando el estándar de programación OpenMP.
* Conocer métricas de desempeño para un código de programación.

**Ejercicios:**

1. Diseñe un programa que permita multiplicar 2 matrices de tamaño 10x10 **usando OpenMP**. Inicie cada matriz con valores constantes dentro del programa. Compruebe el resultado usando un programa alterno (Ejemplo: [Multiplicación de matrices online)](http://es.onlinemschool.com/math/assistance/matrix/multiply/).





1. Use el programa genMatrix para generar matrices de diferente tamaño. El programa recibe como argumento el tamaño de la matriz a generar e imprime en consola la matriz generada. Por ejemplo para generar una matriz de 1000 x 1000 y guardarla en el archivo m1k.txt, ejecute el siguiente comando.

genMatrix 1000 > m1k.txt

1. Modifique el programa para que permita cargar una matriz A, de tamaño mxm, desde un archivo de texto A.txt y retorne el resultado de la multiplicación en otro archivo C.txt. En este caso realice la operación C=AxA. El formato de matriz separa cada columna con tabulación y cada fila con un salto de línea (ver figura 1). El resultado debe ser impreso en este mismo formato.



Figura 1. Ejemplo de una matriz 3x3 en el formato indicado.

1. Divida su código en funciones y determine el tiempo que toma cada función para multiplicar una matriz de tamaño 1000x1000 (ejemplo: Cargar la matriz, multiplicar la matriz, imprimir el resultado), use una ejecución secuencial (1 solo thread). Complete la tabla 1.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **Tiempo** |  |
| Cargar la matriz (F1) | Multiplicar la matriz (F2) | Imprimir el resultado (F3) |
| 1000 x 1000 | 4 segundos | 9 segundos | 3 segundos |

Tabla 1. Tiempos de ejecución para una matriz de tamaño 1000 x 1000.

1. Use diferentes tamaños de la matriz A y calcule el tiempo requerido por su programa para computar la multiplicación C=AxA. En este caso ejecute el programa con diferentes trheads. Finalmente, complete la tabla 2.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| MATRIZ | |  | | P1 | |  | |  | | P2 | |  | |  | | P4 | |  | |
|  | | C | | M | | I | | C | | M | | I | | C | | M | | I | |
|  | | **F1** | | **F2** | | **F3** | | **F1** | | **F2** | | **F3** | | **F1** | | **F2** | | **F3** | |
| 1K X 1K | | 4 | | 9 | | 3 | | 3 | | 6 | | 3 | | 4 | | 4 | | 3 | |
| 2K X 2K | | 4 | | 67 | | 4 | | 3 | | 37 | | 4 | | 4 | | 19 | | 5 | |
| 3K X 3K | | 5 | | 270 | | 6 | | 5 | | 134 | | 6 | | 5 | | 71 | | 6 | |
| 4K X 4K | | 6 | | 594 | | 8 | | 6 | | 296 | | 9 | | 6 | | 150 | | 8 | |
| 5K X 5K | | 9 | | 1285 | | 2 | | 8 | | 627 | | 12 | | 8 | | 327 | | 11 | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| MATRIZ | |  | | P8 | |  | |  | | P16 | |  | |  | | P32 | |  | |
|  | | C | | M | | I | | C | | M | | I | | C | | M | | I | |
|  | | **F1** | | **F2** | | **F3** | | **F1** | | **F2** | | **F3** | | **F1** | | **F2** | | **F3** | |
| 1K X 1K | | 3 | | 4 | | 4 | | 3 | | 4 | | 4 | | 3 | | 4 | | 3 | |
| 2K X 2K | | 4 | | 15 | | 4 | | 3 | | 16 | | 4 | | 3 | | 15 | | 5 | |
| 3K X 3K | | 5 | | 55 | | 6 | | 5 | | 60 | | 5 | | 5 | | 55 | | 6 | |
| 4K X 4K | | 6 | | 125 | | 8 | | 6 | | 122 | | 9 | | 6 | | 123 | | 8 | |
| 5K X 5K | | 8 | | 270 | | 11 | | 9 | | 264 | | 11 | | 9 | | 268 | | 9 | |

F1: Cargar la matriz, F2: Multiplicar la matriz, F3: Imprimir el resultado.

Tabla 2. Tiempos de ejecución al incluir *P* unidades de procesamiento.

1. Use la tabla 2 para calcular el tiempo total de ejecución, el sobrecosto por unidad de procesamiento, el sepeedup local, el speedup global del programa, la escalabilidad, la eficiencia y el costo.

**P= N procesadores**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Tiempo total de ejecución total** | | | | | |
| P=1 | P=2 | P=4 | P=8 | P=16 | P=32 |
| 1000 x 1000 | 16 | 12 | 11 | 11 | 11 | 10 |
| 2000 x 2000 | 75 | 44 | 28 | 23 | 23 | 23 |
| 3000 x 3000 | 281 | 145 | 82 | 66 | 70 | 66 |
| 4000 x 4000 | 608 | 311 | 164 | 139 | 137 | 137 |
| 5000 x 5000 | 1296 | 647 | 346 | 289 | 284 | 286 |

**Costo**

****Costo = Tiempo paralelo x Numero de Núcleos

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Costo** | | | | |
| P=2 | P=4 | P=8 | P=16 | P=32 |
| 1000 x 1000 | 6\*2=12 | 4\*4=16 | 4\*8=32 | 4\*16=64 | 4\*32=128 |
| 2000 x 2000 | 37\*2=74 | 19\*4=76 | 15\*8=120 | 16\*16=256 | 15\*32=480 |
| 3000 x 3000 | 134\*2=268 | 71\*4=284 | 55\*8=440 | 60\*16=960 | 55\*32=1.760 |
| 4000 x 4000 | 296\*2=592 | 150\*4=600 | 125\*8=1000 | 122\*16=1.952 | 123\*32=3.936 |
| 5000 x 5000 | 627\*2=1.254 | 327\*4=1.308 | 270\*8=2.160 | 264\*16=4.224 | 268\*32=8.576 |

1. Determine el porcentaje de su código que puede ser paralizado y use la ley de Amdahl para calcular el speedup máximo. ¿Cuál es la relación entre el speedup teórico y el experimental?

**Amdahl’s Law Speedup**

Texto

Descripción generada automáticamente con confianza baja

Fracción Paralela P=1/3

Fracción Serial S=⅔

|  |  |
| --- | --- |
| N | Speedup |
| 1 |  |
| 2 |  |
| 4 |  |
| 8 |  |
| 16 |  |
| 32 |  |

**Eficiencia**

**Texto, Carta

Descripción generada automáticamente**

|  |  |
| --- | --- |
| Nº | Eficiencia |
| 1 |  |
| 2 |  |
| 4 |  |
| 8 |  |
| 16 |  |
| 32 |  |

**Conclusión:**