Universidad del Valle de Guatemala
Sección 10 - Computación Paralela y Distribuida
Catedrático: Juan Jose Celada

Proyecto 1

Cristopher Jose Rodolfo Barrios Solis - 18207
Juan Fernando de Leon Quezada - 17822
Jose Amado Garcia Rosales - 181469

Índice

Introducción **3**

Antecedentes **4**

OpenMP 4

PCAM 5

Ecuación unidimensional de disipación de calor. 6

Diagrama de flujo **7**

Resultados **8**

Tiempo 8

Funciones Desarrolladas 8

Conclusiones **9**

Recomendaciones **9**

Apéndice **9**

Capturas de Pantalla 9

Introducción

Durante la visualización del documento se podrá encontrar la la utilización de OpenMP para poder paralelizar ciertos bloques de código ya que esta herramienta permite la transformación de un programa secuencial a paralelo, con esto se pueden hacer cambios de una manera que no nos dificulte aprovechar múltiples recursos durante la ejecución.

Se tiene como objetivo aplicar el método PCAMP y los conceptos de patrones de partición asimismo se estará utilizando la ecuación unidimensional de disipación del calor, con el propósito de poder utilizar el programa de secuencial a paralelo, se están tomando varias mediciones de tiempo, y tambien se estara usando el calculo speedup.

Antecedentes

OpenMP

Cuando hablamos de OpenMP podemos decir que es un conjunto de directivas de compilación, así como una API para programas escritos en C, C++ que brinda soporte para programación paralela en entornos de memoria compartida OpenMP identifica regiones paralelas como bloques de código que pueden ejecutarse en paralelo (Chakraborty, 2019).

Cuando se desarrollan las aplicaciones se instalan directivas de compilación en su código en regiones paralelas y estas directivas indican a la biblioteca de tiempo de ejecución de OpenMP que ejecute la región en paralelo (Chakraborty, 2019).

Cuando OpenMP encuentra la directiva:

#pragma omp parallel

Crear tantos subprocesos que procesan núcleos en el sistema. Así para un sistema de doble núcleo, se crean dos subprocesos, para un sistema de cuatro núcleos, se crean cuatro. Luego todos los hilos ejecutan simultáneamente la región paralela. Cuando cada subproceso sale de la región paralela, se termina. OpenMP proporciona varias directivas adicionales para ejecutar regiones de código en paralelo, incluidos los bucles de paralelización (Chakraborty, 2019)..

Permite también a los desarrolladores a elegir entre varios niveles de paralelismo, se pueden establecer el número de subprocesos manualmente. También permite a los desarrolladores identificar si los datos se comparten entre subprocesos o son privados para un subproceso (Chakraborty, 2019).

PCAM

La metodología de Foster tiene cuatro pasos para diseñar algoritmos paralelos, estos son task que interactúan unas con otras:

- Fraccionamiento
    - Se realizan los datos operados por este cálculo y se descomponen en pequeñas tareas. Cuestiones prácticas tales como el número de procesadores en el equipo destino son ignorados y la atención se centra en reconocer oportunidades de ejecución paralela (Foster, 1995).

- Comunicación
    - esta es necesaria para coordinar la ejecución de la tarea es determinado y apropiado estructuras de comunicación y se definen los algoritmos (Foster, 1995).

- Aglomeración
    - La tarea y estructuras de comunicación definidas en las primeras dos etapas de un diseño son evaluadas con respecto al desempeño requisitos y costos de implementación. Si es necesario, las tareas se combinan en tareas más grandes para mejorar el rendimiento o para reducir los costos de desarrollo (Foster, 1995).

- Cartografía
    - Cada tarea se asigna a un procesador de una manera que intente satisfacer los objetivos competitivos de maximizar la utilización de procesador y minimizando los costos de comunicación. El mapeo se puede especificar de forma estática o determinado en tiempo de ejecución por algoritmos de equilibrio de carga (Foster, 1995).

Ecuación unidimensional de disipación de calor.

Con respecto a la ley de fourier se puede decir que la transferencia de calor se puede llegar a cuantificar en términos de ecuaciones apropiadas, en este caso se tiene la velocidad para cuantificar esto se basa en la ley de conducción térmica de Fourier, esta establece la tasa de tiempo de transferencia de calor a través de un material es proporcional al gradiente negativo en la temperatura y al área, en ángulo recto a ese gradiente, a través del cual fluye el calor. La siguiente ecuación se obtiene del principio de la conservación de la energía y de la ley anteriormente explicada PDE(Hector, 2008):

$$\frac{\partial T(x,t)}{\partial t} = c\frac{\partial^2 T(x,t)}{\partial x^2}$$

Esto es para una barra de longitud L, T(x,t) es la temperatura desconocida en la distancia x en el tiempo t y c es la difusividad térmica del material con su valor típico para encontrar la solución simplificada de la PDE, necesitamos la temperatura inicial de la barra T_0 y las temperatura en ambos extremos de la barra T_l y T_R ambas independientes de tiempo. las temperaturas son constantes (Hector, 2008).

Se puede modificar la ecuación anterior utilizando unos intervalos de tiempo, utilizando aproximaciones diferenciales finitas de las derivadas en el tiempo y la distancia se tienen las siguientes aproximaciones (Hector, 2008).

$$\frac{\partial T(x_j,t)}{\partial t} = \frac{T_j(t_{i+1}) - T_j(t_i)}{\Delta t}$$

$$\frac{\partial^2 T(x,t)}{\partial x^2} = \frac{T_{j-1}(t_i) - 2T_j(t_i) + T_{j+1}(t_i)}{(\Delta x)^2},$$

Ahora se utiliza el método explícito de Euler para obtener el siguiente algoritmo para el cálculo de la nueva temperatura (Hector, 2008).

$$T_j(t_{i+1}) = T_j(t_i) + \frac{c\,\Delta t}{(\Delta x)^2}\left(T_{j-1}(t_i) - 2T_j(t_i) + T_{j+1}(t_i)\right).$$

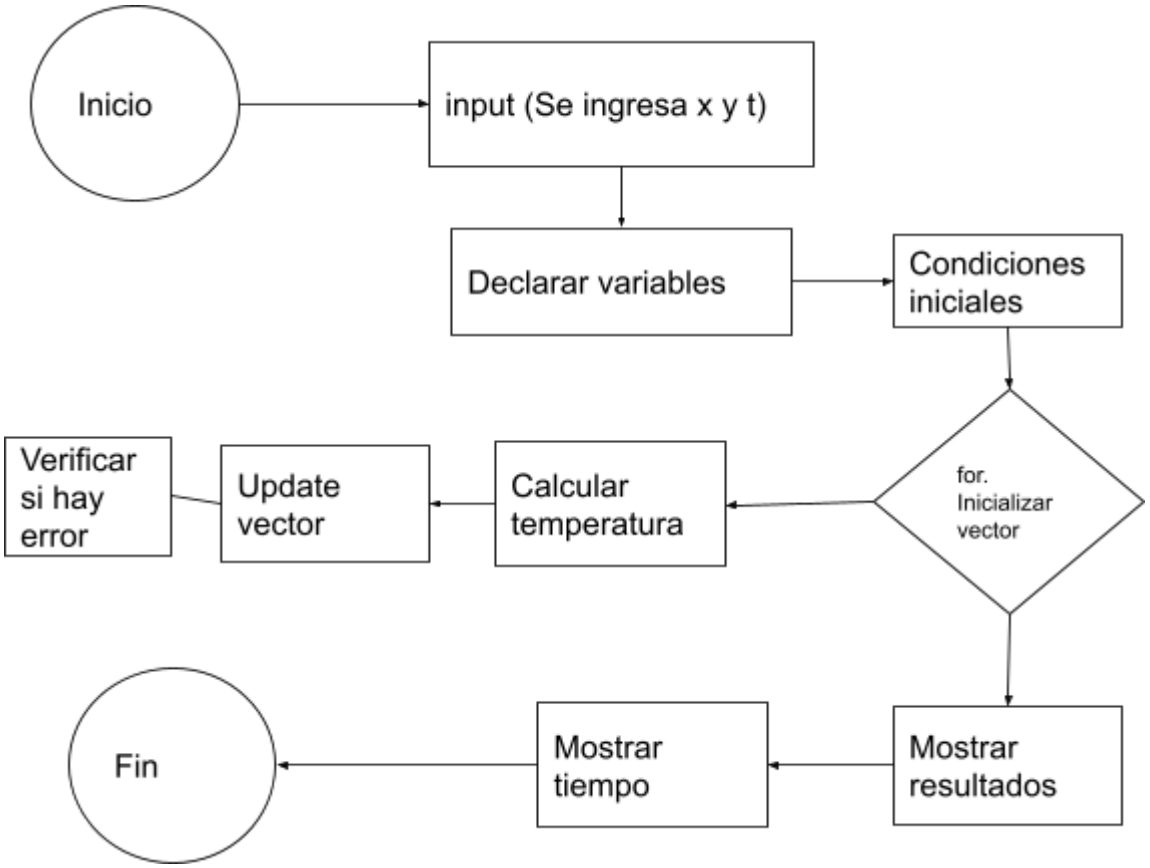
Se obtiene una nueva temperatura anterior y el producto de un factor constante C con la combinación lineal de las tres temperaturas (Hector, 2008).

Constante C:

$$\frac{c\,\Delta t}{(\Delta x)^2}$$

c: difusividad térmica del material
Δt: intervalo de tiempo
Δx: intervalo discreto de la distancia

Diagrama de flujo



Resultados

Tiempo

Cuando ponemos en ejecución el programa se realizan un total de 20 ejecuciones con diferentes datos, como resultado final tenemos el tiempo de ejecución tanto del algoritmo secuencial y paralela en esta tabla se muestran un par de ejecuciones que se obtuvieron.

número	Paralelo (seg)	Secuencial (seg)
1	3.389	0.647
2	3.407	0.666
3	3.425	0.665
4	3.447	0.661
5	3.493	0.662
6	3.382	0.665
7	3.403	0.657
8	3.403	0.663
9	3.396	0.664
10	3.445	0.674
11	3.482	0.654
12	3.457	0.656
13	3.457	0.662
14	3.376	0.653
15	3.485	0.651

Funciones Desarrolladas

double Tj: Ecuación de temperatura, los valores los obtiene según lo ingresa el usuario.
double heatDisipationParallel: Calculo de la temperatura utilizando paralelismo.

Conclusiones

- Ha sido un reto el entendimiento de cómo se utiliza el paralelismo con estas nuevas herramientas, ya que puede llegar a ser complejo por el uso de ecuaciones de física, asimismo en la programación pueden haber algunas cosas que no se entiendan.
- En este caso el tiempo del algoritmo que es paralelo es más rápido como se previó desde el principio ya que el secuencial hace todo de una manera más lineal, en cambio el otro opera con varias ecuaciones al mismo tiempo lo que provoca que el tiempo en la respuesta final sea mas rapida.

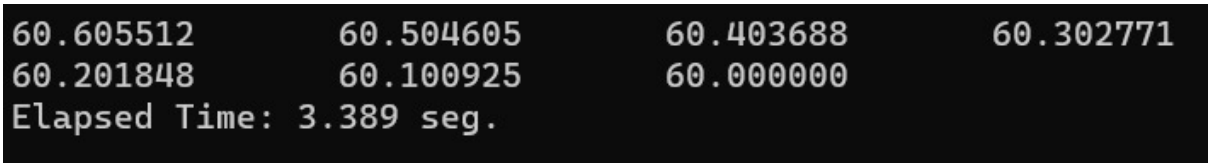
Recomendaciones

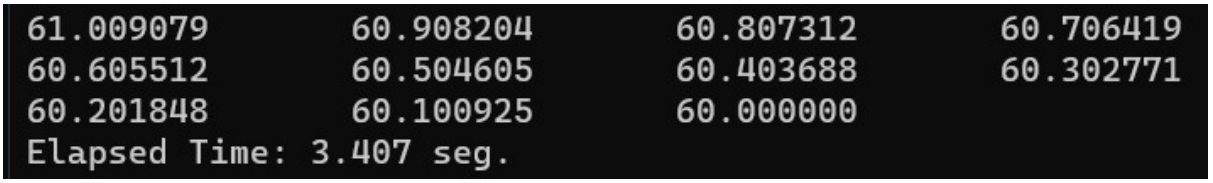
- Se puede encontrar alguna otra ecuación que permita que el tiempo de ejecución sea menor, no solo en la programación se puede acortar el tiempo sino también en el planteamiento de las fórmulas matemáticas que se utilizan para llegar a la respuesta.
- Investigar diferentes formas de paralelismo para obtener mejores resultados, esto puede ampliar el conocimiento que se tiene sobre el tema y no solo limitarse a una sola manera de resolver el problema.

Apéndice

Capturas de Pantalla

Algoritmo secuencial





```
61.009079       60.908204       60.807312       60.706419
60.605512       60.504605       60.403688       60.302771
60.201848       60.100925       60.000000
Elapsed Time: 3.493 seg.
```

```
60.605512       60.504605       60.403688       60.302771
60.201848       60.100925       60.000000
Elapsed Time: 3.382 seg.
```

```
60.605512       60.504605       60.403688       60.302771
60.201848       60.100925       60.000000
Elapsed Time: 3.403 seg.
```

```
60.605512       60.504605       60.403688       60.302771
60.201848       60.100925       60.000000
Elapsed Time: 3.403 seg.
```

```
60.605512       60.504605       60.403688       60.302771
60.201848       60.100925       60.000000
Elapsed Time: 3.396 seg.
```

```
60.605512       60.504605       60.403688       60.302771
60.201848       60.100925       60.000000
Elapsed Time: 3.445 seg.
```

```
60.605512       60.504605       60.403688       60.302771
60.201848       60.100925       60.000000
Elapsed Time: 3.482 seg.
```

```
60.605512       60.504605       60.403688       60.302771
60.201848       60.100925       60.000000
Elapsed Time: 3.457 seg.
```

```
60.605512       60.504605       60.403688       60.302771
60.201848       60.100925       60.000000
Elapsed Time: 3.457 seg.
```

```
60.605512       60.504605       60.403688       60.302771
60.201848       60.100925       60.000000
Elapsed Time: 3.376 seg.
```

```
60.605512       60.504605       60.403688       60.302771
60.201848       60.100925       60.000000
Elapsed Time: 3.485 seg.
```

Algoritmo paralelo

```
C:\Users\jfdel\Desktop\UVG2022-1\Computacion Paralela y Distribuida\Proyecto-1\Proyecto-1-CC3069-OpenMP>main.exe
Elapsed Time: 0.647 seg.
```

```
C:\Users\jfdel\Desktop\UVG2022-1\Computacion Paralela y Distribuida\Proyecto-1\Proyecto-1-CC3069-OpenMP>main.exe
Elapsed Time: 0.666 seg.
```

```
C:\Users\jfdel\Desktop\UVG2022-1\Computacion Paralela y Distribuida\Proyecto-1\Proyecto-1-CC3069-OpenMP>main.exe
Elapsed Time: 0.665 seg.
```

```
C:\Users\jfdel\Desktop\UVG2022-1\Computacion Paralela y Distribuida\Proyecto-1\Proyecto-1-CC3069-OpenMP>main.exe
Elapsed Time: 0.661 seg.
```

```
C:\Users\jfdel\Desktop\UVG2022-1\Computacion Paralela y Distribuida\Proyecto-1\Proyecto-1-CC3069-OpenMP>main.exe
Elapsed Time: 0.662 seg.
```

```
C:\Users\jfdel\Desktop\UVG2022-1\Computacion Paralela y Distribuida\Proyecto-1\Proyecto-1-CC3069-OpenMP>main.exe
Elapsed Time: 0.665 seg.
```

```
C:\Users\jfdel\Desktop\UVG2022-1\Computacion Paralela y Distribuida\Proyecto-1\Proyecto-1-CC3069-OpenMP>main.exe
Elapsed Time: 0.657 seg.
```

```
C:\Users\jfdel\Desktop\UVG2022-1\Computacion Paralela y Distribuida\Proyecto-1\Proyecto-1-CC3069-OpenMP>main.exe
Elapsed Time: 0.663 seg.
```

```
C:\Users\jfdel\Desktop\UVG2022-1\Computacion Paralela y Distribuida\Proyecto-1\Proyecto-1-CC3069-OpenMP>main.exe
Elapsed Time: 0.664 seg.
```

```
C:\Users\jfdel\Desktop\UVG2022-1\Computacion Paralela y Distribuida\Proyecto-1\Proyecto-1-CC3069-OpenMP>main.exe
Elapsed Time: 0.674 seg.
```

```
C:\Users\jfdel\Desktop\UVG2022-1\Computacion Paralela y Distribuida\Proyecto-1\Proyecto-1-CC3069-OpenMP>main.exe
Elapsed Time: 0.654 seg.
```

```
C:\Users\jfdel\Desktop\UVG2022-1\Computacion Paralela y Distribuida\Proyecto-1\Proyecto-1-CC3069-OpenMP>main.exe
Elapsed Time: 0.656 seg.
```

```
C:\Users\jfdel\Desktop\UVG2022-1\Computacion Paralela y Distribuida\Proyecto-1\Proyecto-1-CC3069-OpenMP>main.exe
Elapsed Time: 0.662 seg.
```

```
C:\Users\jfdel\Desktop\UVG2022-1\Computacion Paralela y Distribuida\Proyecto-1\Proyecto-1-CC3069-OpenMP>main.exe
Elapsed Time: 0.653 seg.
```

```
C:\Users\jfdel\Desktop\UVG2022-1\Computacion Paralela y Distribuida\Proyecto-1\Proyecto-1-CC3069-OpenMP>main.exe
Elapsed Time: 0.651 seg.
```

Bibliografía

- Hector, L. (2008). Aplicaciones de las Ecuaciones Diferenciales Parciales. Extraído de: http://sgpwe.izt.uam.mx/files/users/uami/hect/EDP/presentacion.pdf
- Chakraborty, A. (2019). What is OpenMP? Extraido de: https://www.tutorialspoint.com/what-is-openmp
- Foster, L. (1995). Parallel Algorithm Analysis and Design. Extraido de: https://www.math-cs.gordon.edu/courses/cps343/presentations/Parallel_Alg_Design.pdf