

Laboratorio 1

Nombre: Raúl Alejandro Monzon Solís

Carné: 17014

Competencias a desarrollar


Comprende los fundamentos de computación paralela, distribuida, tipos de paralelismo y sus efectos en la programación, por medio de ejercicios prácticos resueltos en parejas. Debe aplicar y probar el uso del Método de Foster y las estrategias de partición de tareas.

Instrucciones

Ejecute cada una de las actividades y anote sus resultados. Recuerde resumir sus observaciones y discutir los resultados.

Actividades

1. **(5 puntos).** Compile y ejecute el código `trapOMP1.c`. Ejecute el programa para los valores: $a=0$, $b=16$; $n=8$ y ejecute con el mismo número de hilos que núcleos de su computadora.

 C:\Users\User\Downloads\trapOMP1.exe

```
Enter a, b, and n
0
16
8
With n = 8 trapezoids, our estimate
of the integral from 0.000000 to 16.000000 = 1.376000000000000e+003

Process returned 0 (0x0)   execution time : 19.368 s
Press any key to continue.
```

2. **(5 puntos)** Vuelva a correr el programa, ahora modificando el número de trapezoides $n = [10, 12, 14, 16, 20]$ y manteniendo el número de hilos al número de núcleos de sus sistema. Anote sus resultados

Valor de n	Tiempo
10	6.810 s
12	3.589 s
14	4.127 s
16	3.168 s
20	3.006 s

3. (5 puntos) Ejecute nuevamente el programa para los valores: $a=0$, $b=16$, $n=32$. Modifique el número de hilos a ejecutar en múltiplos de k núcleos: $2k$, $4k$, $8k$, $16k$. Registre los resultados.

Hilos	Integral
16	1.3660000000000000e+003
32	1.3660000000000000e+003
64	0.0000000000000000e+000
128	0.0000000000000000e+000

4. (10 puntos) La división del rango local dentro del número local de trapecoides puede dar un número no entero. Esto presenta problemas de precisión al momento de calcular las sumas locales. Modifique el programa original para que cada hilo reciba un número entero de trapecoides.

Se encuentra en el repo de github

5. (15 puntos) Modifique el programa original para eliminar la sección crítica y aplicar una operación de reducción. Compare con los resultados anteriores usando 3 modificaciones al número de hilos y 3 modificaciones al número de trapecoides a su elección.

Se encuentra en el repo de github

6. (15 puntos) La variable factor es global pues está creada antes de la directiva parallel for. Esto ocasiona que todos los hilos tengan acceso a modificar esta variable y así afectar el flujo de ejecución de otro hilo. OpenMP nos permite hacer una declaración explícita de la privacidad de una variable, para que cada hilo obtenga una copia de la misma, mediante la cláusula private(variable) dentro de la directiva parallel for. Modifique su programa y ejecútelo nuevamente, comparando los resultados de esta nueva versión con la anterior para cálculo de PI.

Se encuentra en el repo de github

7. (10 puntos) INDIVIDUAL Discuta sus resultados para:

- Lo que sucede cuando modifica el número de trapecoides. ¿Cuáles pueden ser las razones?
Por la cantidad de hilos que van variando
- ¿Qué sucede cuando modifica el número de núcleos?
El tiempo de ejecución reduce.



Universidad del Valle de Guatemala
Facultad de Ingeniería
Departamento de Ciencias de la Computación
CC3069 Computación Paralela y Distribuida

Ciclo 1 de 2,021