Universidad del Valle de Guatemala
Facultad de Ingeniería
Departamento de Ciencias de la Computación
CC3069 Computación Paralela y Distribuida Ciclo 1 de 2,021

Laboratorio 1

Nombre: Carlos Chew Carné: 17507

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. Registre los resultados en una tabla.

A	**0**
B	16
N	8
THREADS	8
INT	1.37600000000000e+003

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

A	**0**	**0**	**0**	**0**	**0**
B	16	16	16	16	16
N	10	12	14	16	20
THREADS	8	8	8	8	8
INT	7.04512000000000e+002	4.07703703703704e+002	2.56746355685131e+002	1.36800000000000e+003	7.00416000000000e+002

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.

A	*0*	*0*	*0*	*0*
B	16	16	16	16
N	32	***32***	***32***	***32***
THREADS	***16***	***32***	***64***	***128***
INT	1.36600000000000e+003	1.36600000000000e+003	***0***	***0***

4. **(10 puntos)** La división del rango local dentro del número local de trapezoides 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 trapezoides.

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 trapezoides a su elección. Registre sus resultados.**

6. **(15 puntos)** En el programa `ompPInaive.c`, introdujimos una dependencia de loop, ya que la variable `factor` depende del valor anterior de iteración. Debido a que OpenMP planifica los hilos, no tenemos garantía que el factor en una iteración esté correcto, pues su valor anterior puede estar asignado al dominio de otro hilo. Modifique el programa para que `factor` no dependa del valor anterior, sino que esté relacionado con el dominio de la variable k asignado a cada hilo. Ejecute el programa con cualquier valor de hilos (2, 4, 8) y con un número de iteraciones superior a 50,000. Registre 6 resultados.

7. **(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.**

8. **(10 puntos)** INDIVIDUAL Discuta sus resultados para:
   a. Lo que sucede cuando modifica el número de trapezoides. ¿Cuáles pueden ser las razones?
      i. Se asigna más trabajo a cada hilo.
   b. ¿Qué sucede cuando modifica el número de núcleos?
      i. El resultado evidentemente es diferente, ya que se asigna más capacidad o menos capacidad.
   c. Discuta los resultados en tiempo de ambas versiones. Justifique. Que riesgos y ventajas puede tener cada forma de dividir el trabajo en la sección paralela.
      i. Cuando son muchos calculos trabajar con paralelismo es muy eficiente.
   d. ¿Cómo vemos los efectos de cambiar la dependencia y scope de la variable *factor* en el cálculo de PI?
      i. Es más exacto al calcular el valor de pi