# Laboratorio 1

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre: | Marlon Fuentes | Carné: | 15240 |

**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.

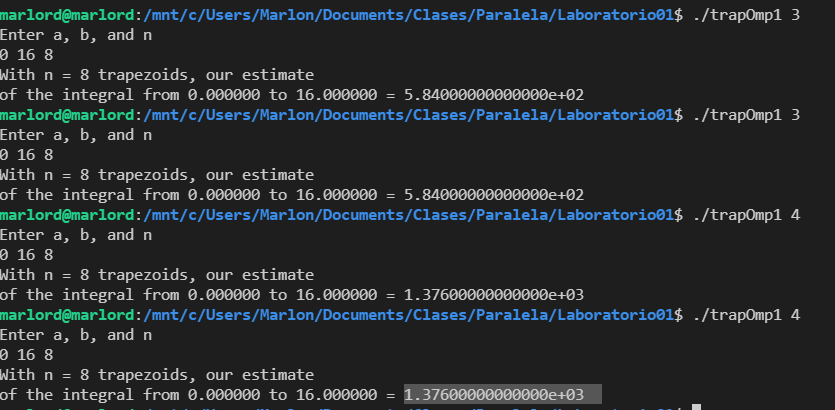
**Instrucciones**

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

***Actividades***

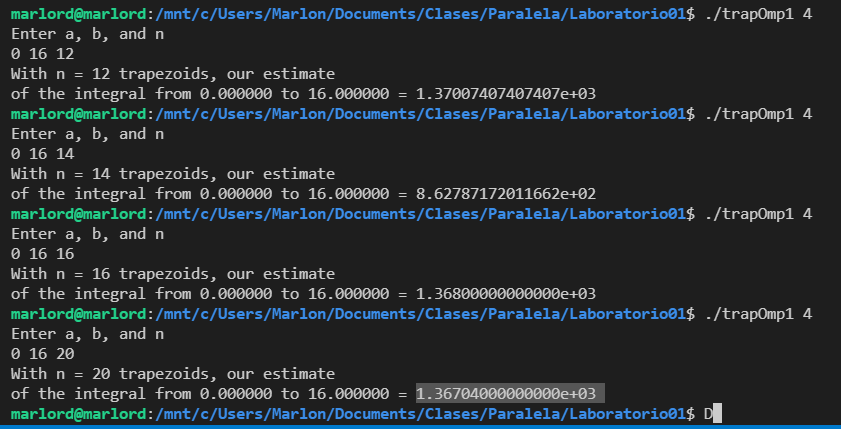
1. **(5 puntos).** Compile y ejecute el código trapOMP1.c Ejecute el programa para los valores: a=0, b=16; n=8 y ejecute con el mismo número de hilos que núcleos de su computadora. Registre los resultados en una tabla.

|  |  |
| --- | --- |
| Núcleos | Resultado |
| 1 | 1.37600000000000e+03 |
| 2 | 1.37600000000000e+03 |
| 3 | 5.84000000000000e+02 |
| 4 | 1.37600000000000e+03 |



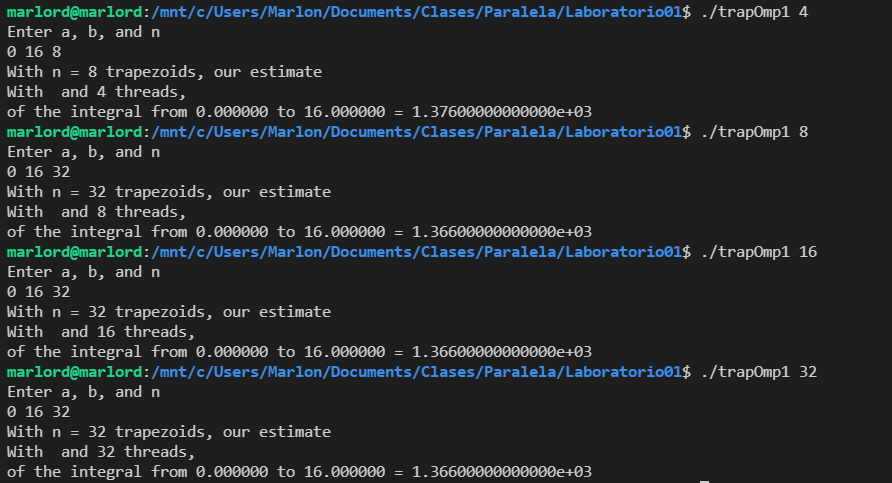
1. **(5 puntos)** Vuelva a correr el programa, ahora modificando el número de trapezoides n = [10, 12, 14, 16, 20] y manteniendo el número de hilos al número de núcleos de sus sistema. Anote sus resultados

|  |  |
| --- | --- |
| Trapezoides | Resultado |
| 10 | 7.04512000000000e+02 |
| 12 | 1.37007407407407e+03 |
| 14 | 8.62787172011662e+02 |
| 16 | 1.36800000000000e+03 |
| 20 | 1.36704000000000e+03 |

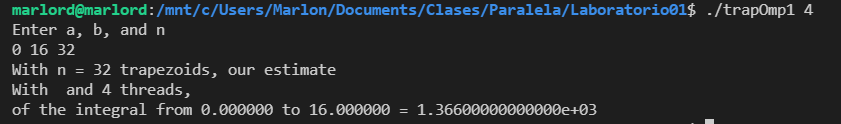


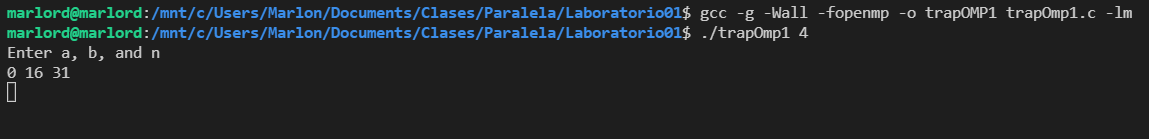
1. **(5 puntos)** Ejecute nuevamente el programa para los valores: a=0, b=16, n=32. Modifique el número de hilos a ejecutar en múltiplos de k núcleos: 2k, 4k, 8k, 16k. Registre los resultados.

|  |  |
| --- | --- |
| **Núcleos** | **Resultado** |
| **4** | **1.37600000000000e+03** |
| **8** | **1.36600000000000e+03** |
| **16** | **1.36600000000000e+03** |
| **32** | **1.36600000000000e+03** |



1. **(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.





El programa ya no corre si la división no regresa un par

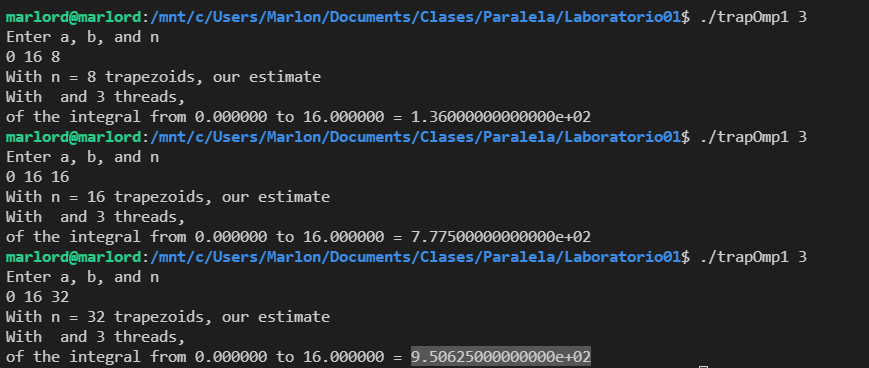
1. **(15 puntos) Modifique el programa original para eliminar la sección crítica y aplicar una operación de reducción. Compare con los resultados anteriores usando 3 modificaciones al número de hilos y 3 modificaciones al número de trapezoides a su elección. Registre sus resultados.**

**N=32**

|  |  |
| --- | --- |
| **Núcleos** | **Resultados** |
| **1** | **1.23800000000000e+03** |
| **2** | **1.20600000000000e+03** |
| **3** | **9.50625000000000e+02** |

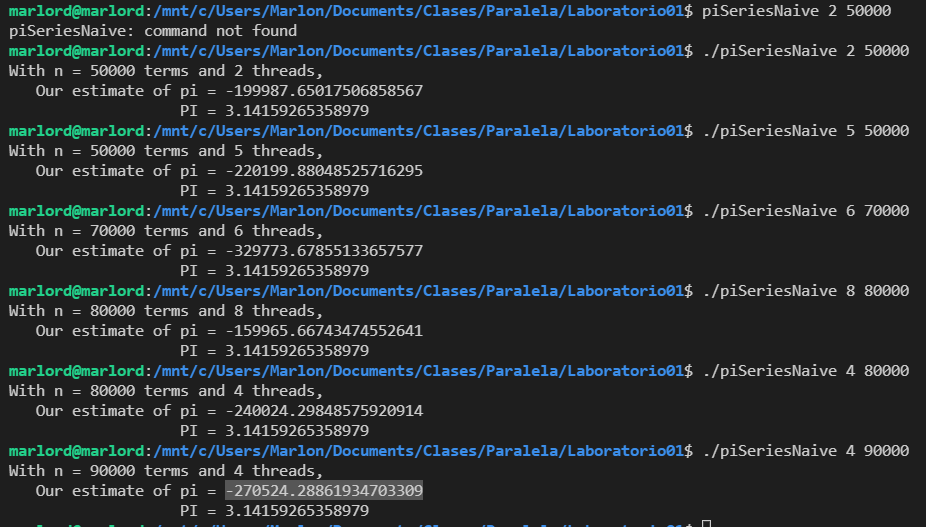
**N=3**

|  |  |
| --- | --- |
| **Trapezoides** | **Resultados** |
| **8** | **1.36000000000000e+02** |
| **16** | **7.77500000000000e+02** |
| **32** | **9.50625000000000e+02** |



1. **(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.

|  |  |
| --- | --- |
| Hilo | Resultado |
| 2 | -199987.65017506858567 |
| 6 | -220199.88048525716295 |
| 8 | -329773.67855133657577 |
| 4 | -159965.66743474552641 |
| 2 | -240024.29848575920914 |
| 4 | -270524.28861934703309 |



1. **(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.**
2. **(10 puntos)** INDIVIDUAL Discuta sus resultados para:
   1. Lo que sucede cuando modifica el número de trapezoides. ¿Cuáles pueden ser las razones?

Por lo que veo, cuando modificamos los trapezoides se puede ver una variación del resultado. Una de las razones puede ser la complejidad que presenta para los núcleos manejar la cantidad de información que le solicitamos. También el poder de computo es un factor que interactúa bastante con esto. Siento que por mi computadora no tiene mucho poder para manejar threads, entonces muestra resultados similares ya que puede que no este usando los core de forma eficiente.

* 1. ¿Qué sucede cuando modifica el número de núcleos?

Mejora el calculo, para los primeros tres casi no se ve diferencia en los valores que retorna. Cuando ya maneja cuatro núcleos que es lo máximo que tiene mi computadora muestra un valor más preciso.

* 1. 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

Riesgos: El cálculo puede ser menos preciso en la primera versión ya que estamos dependiendo mucho de las variables. La segunda versión esta limitada al poder de computo de la misma. Si es posible puede manejar toda la carga, de lo contrario puede que realice un cálculo a medias.

Ventajas: La segunda versión es más eficiente ya que podemos evitar un cuello de botella. Si la computadora tiene los recursos suficientes, el resultado es más preciso.

* 1. ¿Cómo vemos los efectos de cambiar la dependencia y scope de la variable *factor* en el cálculo de PI?

El factor de Pi si varia bastante ya que la dependencia que tenia previa realizaba un cálculo que si es necesario para el factor. El valor de K es aproximado, pero muestra resultados correctos. Finalmente es más eficiente porque ya no realiza el cálculo que hacía antes.

**Bitácora de Resultados Trapezoide (5 puntos)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Inciso** | **Programa** | **n**  **trapezoides** | **T**  **hilos** | **Test 1** | **Test 2** | **Test 3** |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

**Bitácora de Resultados cálculo de PI (5 puntos)**