

Artículo Proyecto 3

Erick Cordero Rojas, Victor Montero Barboza, Cristófer Villegas Gonzalez
 eguicoro2@gmail.com vic9565@gmail.com mcxcissx@gmail.com
 Área académica de Ingeniería en Computadores
 Análisis Numérico para Ingeniería
 Instituto Tecnológico de Costa Rica

Resumen—Artículo del proyecto 3 del curso Análisis Numérico, enfocado en Ecuaciones Diferenciales Parciales y Splines. También se utilizaron otras herramientas como OpenMP, para la paralelización de procesos y Doxygen para la documentación

Palabras clave—OpenMP, Doxygen, EDP, Splines

I. INTRODUCCIÓN

En este proyecto se realizan cálculos de ecuaciones Diferenciales Parciales, para esto se necesita, de alguna forma poder acelerar los cálculos, ya que estos toman mucho tiempo. Para poder hacer esto es necesario el uso de OpenMP para poder tener procesos ejecutándose de forma simultánea.

Más adelante se analizará el efecto; tanto de la utilización, como de la no utilización de OpenMP respecto a tiempo de ejecución del código.; comparando los tiempos en cada caso

II. ANTECEDENTES

Para este proyecto se da la situación en la que se ejecutan códigos que tienen muchos cálculos, y se iteran en cantidades muy grandes, este problema se ve representado en el tiempo que necesita el proyecto para poder llegar a la solución.

Ante esta necesidad se plantea la solución de utilizar OpenMP para poder realizar cálculos y ejecuciones de secciones de código de forma paralela y de esta manera reducir los tiempos de ejecución.

III. PROPUESTA

Para una utilización de OpenMP de manera correcta se propone utilizar la paralelización en el algoritmo de Liebmann. Para esto se debe diseñar este algoritmo de forma que sea más enfocado a la paralelización. Para esto el algoritmo se diseña de forma que se recorra la matriz en dos mitades, que van a recorrerla y hacer los cálculos de forma simultánea.

Para esto se hacen los cálculos tomando la matriz en dos mitades, la mitad superior y la mitad inferior, para esto se dividen entre 2 los valores de las dimensiones de la matriz. Como este proceso se da de forma paralela el resultado se obtiene de forma más rápida que ejecutar un solo bucle en la matriz entera.

Enfocándose en los procesadores que tenemos en nuestras computadoras, la forma más óptima es hacer una ejecución simultánea de 4 hilos, debido a que tenemos 4 núcleos en el procesador, pero esto no pudo ser implementado por cuestiones de tiempo, así que solo usaremos 2 hilos.

IV. RESULTADOS

Para hacer una comparación entre el código usando OpenMP y sin utilizarlo se va a tomar una matriz de 20x20 y dándole valores de temperatura a todos los extremos de la palca, a partir de esto se va a calcular Liebmann en ambos casos:

■ Caso con OpenMP

```
[100%] Built target benchmark
seconds since start: 35
```

■ Caso sin OpenMP

```
[100%] Built target benchmark
seconds since start: 676
```

Por medio de esta pequeña prueba podemos obtener que:

$$\rho = T2/T1$$

Sustituyendo

$$\rho = 676s/35s$$

Y esto nos da un resultado

$$\rho = 19,314$$

Esto significa que la ejecución con OpenMP, para este caso, es 19.314 veces más rápida que sin OpenMP y en ambas obteniendo el mismo resultado. Conforme la cantidad de iteraciones aumenta esa razón de mejora también lo hace, puesto que ahí es donde más se nota el verdadero potencial de OpenMP.

V. CONCLUSIONES

La utilización de OpenMP provee la capacidad de paralelizar diferentes partes del código para que se ejecuten de forma simultánea y por diferentes hilos del procesador, esto disminuye los tiempos de ejecución y de los cálculos.

Para utilizar OpenMP de forma correcta es necesario proveer su utilización para de esta manera redactar el código en pro de que funciones de la mejor manera con OpenMP.

Si se utiliza OpenMP con ejecuciones muy cortas el proceso es completamente inverso al esperado; ya que en lugar de disminuir el tiempo de ejecución mas bien lo aumenta y no solo aumenta el tiempo de ejecución, sino tambien el uso de los recursos del computador.

REFERENCIAS

- [1] Anonimo. (2011). OPENMP C++ Examples of Parallel Programming with OpenMP. 8/6/2018, de OpenMp Sitio web: https://people.sc.fsu.edu/~jburkardt/cpp_src/openmp/openmp.html