MEMORIA P2 CPA

ISMAEL FERNÁNDEZ HERRERUELA

***EJERCICIO 1***

En este primer ejercicio se nos solicitaba realizar la paralelización de la parte 1 y la parte 3 ya que la 2 no era posible paralelizarla.

Texto

Descripción generada automáticamente

Al realizar la paralelización de la primera parte, utilizamos la **d como variable privada** ya que como con cada hilo se modificaría, no queremos que eso ocurra.

Captura de pantalla con la imagen de una pantalla

Descripción generada automáticamente

En la parte 3 ocurre lo mismo, queremos que **la variable v** se utilice como **privada** para que no se modifique con cada hilo. Y dentro del **else** utilizamos un **pragma omp for** para cada iteración del bucle.

***EJERCICIO 2***

En el segundo ejercicio se nos pedía realizar la paralelización mediante una única región paralela, paralelizar el bucle externo de la primera parte y el bucle de la tercera parte.

Texto

Descripción generada automáticamente

En primer lugar, **privatizamos** las variables **dmin**, **d**, **off**, **bestoff** y **v** para que cada hilo obtenga su **copia local**. Acto seguido **paralelizamos** el bucle **for externo** de la primera parte. A continuación, utilizamos un **pragma omp single** porque debido a que la segunda parte no se puede paralelizar, la ejecutamos una sola vez. Y en ultimo lugar paralelizamos el bucle for de la parte 3.

***EJERCICIO 3***

La planificación Dynamic debería de ofrecer mejores resultados. Esto se debe a como realiza la asignación de hilos en comparación con la static. Si por ejemplo realizamos una planificación static con chunk 4, el primer hilo haría las 4 primeras iteraciones, el segundo hilo haría las 4 siguientes y así sucesivamente, lo cual puede llevarnos a unos tiempos de ejecución mas elevados. Mientras que por otra parte la planificación Dynamic va asignando hilos según se van necesitando, de manera que mientras que una ejecución dura 5 segundos, hay otras 3 ejecuciones que duran 1 segundo cada una, entonces al distribuir los hilos de manera dinámica, si tuviésemos 4 hilos, en un hilo realizaríamos la de 5 y con los otros 3 hariamos las 3 ejecuciones de 1 segundo, por lo que el tiempo total de ejecución seria de 5 segundos ya que es el hilo que mas ha tardado en ejecutarse. En el caso de static, tuviésemos un hilo por cada 4 ejecuciones, tardaría 8 segundos ya que primero haría la de 5 segundos, después una de 1 segundo y así hasta acabar.

***EJERCICIO 4***

**Dynamic default**

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

**Static default**

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

**Static Chunk 1**

Texto

Descripción generada automáticamente

Como podemos observar, la planificación Dynamic es significativamente mas rápida que las demás, esto es debido a que realiza la asignación de hilos en tiempo de ejecución y va asignando según se vayan liberando los hilos de manera automática. Por lo tanto reduce el tiempo de ejecución ya que en lo que acaba la ejecución de un hilo que tarda mas tiempo, se habrán realizado ejecuciones de otros hilos que tarden menos tiempo, a diferencia del static que asigna los hilos sin tener en cuenta el tiempo y esto resulta en un tiempo mayor de ejecución.

***EJERCICIO 5***

**REALIGN 2**

Tiempos con planificación dinámica:

Interfaz de usuario gráfica, Texto, Chat o mensaje de texto

Descripción generada automáticamente

Tiempos con planificación estática:

Texto, Chat o mensaje de texto

Descripción generada automáticamente

Tiempos con planificación estática 1 chunk:

Interfaz de usuario gráfica, Texto, Chat o mensaje de texto

Descripción generada automáticamente

**REALIGN 1**

Tiempos con planificación estática:

**Texto

Descripción generada automáticamente**

Tiempos con planificación estática 1 chunk:

**Texto, Chat o mensaje de texto

Descripción generada automáticamente**

Tiempos con planificación dinámica:

**Texto, Chat o mensaje de texto

Descripción generada automáticamente**

Utilizando en potencias de 2, desde 1 hasta 256 hilos. Como podemos observar, a partir de los 64-128 hilos vuelve a incrementar el tiempo y esto es debido a que se realizan tantas creaciones de hilos y se distribuyen tanto las tareas que aumenta el tiempo. No he realizado pruebas con mas hilos debido a que ya se comprueba que a partir de los 64-128 vuelve a subir el tiempo.

Para calcular el speedup y la eficiencia obtuve el tiempo que tarda sin paralelizar el cual es el siguiente: .

**REALIGN1******

**REALIGN2******

En conclusión, observando todos los resultados, speedups y eficiencias, podemos confirmar que la segunda versión es mas rápida y mas eficiente.

Para lanzarlo en el cluster he utilizado un archivo .sh con el siguiente contenido: Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Esa es solo la primera versión de la paralelización pero para la segunda versión simplemente habría que cambiar ./realign1 por ./realign2 y para cambiar la planificación, en la sentencia de OMP\_SCHEDULE por dynamic, static o static,1.