Diferencias Finitas para resolver la ecuación de calor

1. Objetivos

La realización de esta práctica persigue los siguientes objetivos:

- Familiarizar al alumno con el modelo de programación paralela basado en paso de mensajes (MPI), utilizado en sistemas de memoria distribuida.
- Introducir los conceptos básicos de la programación paralela híbrida en sistemas compuestos por nodos un conjunto de cómputo con múltiples cores cada uno.
- Diseñar e implementar un algoritmo de equilibrio de carga de trabajo para optimizar el rendimiento de las aplicaciones, en sistemas heterogéneos.
- Analizar el rendimiento y la escalabilidad de la aplicación propuesta, tanto en cuanto al escalado fuerte como al débil.

2. Descripción del problema

El calor es una propiedad física que se transmite por convección. De esta forma, si tenemos un espacio cerrado, con zonas a diferentes temperaturas, el calor se irá expandiendo a lo largo del tiempo, hasta igualar la temperatura en dicho espacio. Este proceso está definido mediante la denominada *ecuación del calor* que es una ecuación diferencial en derivadas parciales que describe la distribución del calor (o variaciones de la temperatura) en una región del espacio a lo largo del transcurso del tiempo. La expresión matemática que defina esta ecuación es la siguiente:

$$\frac{\partial T}{\partial t} = \alpha \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} \right) \tag{1}$$

Es una ecuación muy importante en física por lo que se han utilizado múltiples métodos para resolverla. El método numérico más empleado es el Método de Diferencias Finitas (FDM por sus siglas en inglés). Estos métodos resuelven ecuaciones diferenciales mediante la aproximación de derivadas con diferencias finitas. Tanto el dominio espacial como el intervalo de tiempo se discretizan en un número finito de pasos, y el valor de la solución en estos puntos discretos se aproxima resolviendo ecuaciones algebraicas que contienen diferencias finitas y valores de puntos cercanos.

La aproximación utilizada en el caso de la ecuación del calor aplicando el método de diferencias finitas es la siguiente:

$$u_{i,j,k}^{t+1} = \frac{\Delta t}{\Delta h^2}\alpha \begin{bmatrix} u_{i+1,j,k}^{t} + u_{i-1,j,k}^{t} + u_{i,j+1,k}^{t} + u_{i,j-1,k}^{t} \\ +u_{i,j,k+1}^{t} + u_{i,j,k-1}^{t} - 6 \times u_{i,j,k}^{t} \end{bmatrix} - u_{i,j,k}^{t+1}$$

El algoritmo es un proceso iterativo que va convergiendo hasta encontrar la solución con un error menor que un umbral predefinido. En general, cuando más fina es la discretización del espacio (es decir más divisiones se realizan) será necesario un menor número de pasos para la resolución.

3. Material Necesario

Se proporciona un fichero comprimido `fdm.c`, mediante la plataforma Moodle. Este programa implementa el algoritmo de Diferencias Finitas para resolver la ecuación del calor en un espacio a lo largo del tiempo, incluyendo la generación de los datos de entrada, así como la escritura de los datos de salida.

El programa `fdm.c` contiene las funciones necesarias para resolver la ecuación diferencial del calor, mediante el método de diferencias finitas. Incluye la construcción de las estructuras de datos necesarias, la reserva de memoria y la generación aleatoria de los datos de calor en un espacio cerrado concreto, dentro del programa principal.

Los datos de entrada de este programa se proporcionan a través de la línea de comandos. Solamente es necesario proporcionar un dato de entrada que es la variable *deltaH* que indica el número de divisiones que se realizarán en cada dimensión del espacio (x, y, z). Éste es un valor en coma flotante que debe ser menor que 1.0. Cuanto más pequeño sea, más divisiones se realizarán y más costoso será el cómputo. Al ser divisiones en las tres dimensiones del espacio la complejidad del algoritmo es $O(deltaH^3)$.

Además del programa principal sólo se incluya una función `mdf_heat`, que es la encargada de realizar todo el proceso, y por lo tanto la función que se debe paralelizar.

La salida del programa consta únicamente de un valor entero, que corresponde con el número de pasos que ha sido necesario realizar para que el error cometido por el método sea menor que un cierto umbral predefinido.

4. Desarrollo

Ejercicio 1

Realiza la mejor paralelización del código posible utilizando el modelo de programación paralela de paso de mensajes, MPI. Trabajaremos sobre un sistema compuesto por cuatro nodos de cómputo, por lo que será necesario lanzar cuatro procesos. Comprueba que el

resultado del programa sea correcto, comparando los resultados con los obtenidos en el programa secuencial. Mide el tiempo de ejecución del programa secuencial y del paralelo. Calcula el speedup y la eficiencia obtenida y explica los resultados.

Ejercicio 2

Consideramos ahora que cada uno de los nodos de cómputo sobre los que trabajamos es un sistema compuesto por 4 procesadores, sin multi-threading. Para explotar esta arquitectura es necesario utilizar programación paralela híbrida, mezclando los modelos de paso de mensajes y memoria compartida. Por lo tanto, es necesario paralelizar cada uno de los procesos MPI obtenidos en el ejercicio anterior con directivas de OpenMP. Realiza la mejor paralelización posible, añadiendo al código las directivas y/o funciones de OpenMP necesarias. Etiqueta todas las variables y explica por qué le has asignado esa etiqueta. Calcula el speedup y la eficiencia obtenida y explica los resultados.

Ejercicio 3

Implementar un algoritmo de equilibrio de carga de trabajo que sea capaz de distribuir los datos entre los procesos, de forma que cada uno tenga una cantidad de trabajo proporcinal a su potencia de cómputo. El objetivo de este algoritmo es que todos los procesos finalicen su ejecución aproximadamente a la vez. El algoritmo será centralizado y debe tener en cuenta la heterogeneidad del sistema.

Ejercicio 4

Determina, hasta dónde sea posible con el computador con el que estás trabajando, el escalado fuerte y el escalado débil de esta aplicación. Explica las pruebas que has realizado para ello. Extrae todas las conclusiones que puedas sobre el comportamiento de este programa.

5. Evaluación

La evaluación de esta práctica se realizará mediante una presentación del trabajo realizado en el aula, el día que se indique en el Moodle de la asignatura. La duración de la presentación será de 10 minutos. Además se deberá entregar dicha presentación, así como el código realizado, mediante una tarea de la plataforma Moodle.