

Computación Paralela

Profesor Responsable: Héctor Fco Migallón Gomis hmigallon@umh.es			
Profesor de Laboratorio: Héctor Fco Migallón Gomis hmigallon@umh.es			
Departamento: FÍSICA Y ARQUITECTURA DE COMPUTADORES			
Área de Conocimiento: Arquitectura y Tecnología de Computadores			
Curso: 3º	Docencia: 1Sem.	Tipo: Obligatoria	Créditos: 6 ECTS (60 + 90 horas)
Página web de la asignatura: (institucional)			

• **PRACTICA 2**

El objetivo de esta práctica es aprender a programar con el paradigma OpenMP para sistemas de memoria compartida. Se deben realizar dos tareas y analizar su comportamiento mediante ejecuciones de prueba. Los programas a realizar deben implementar las siguientes operaciones en una arquitectura paralela de memoria compartida.

- Esquema iterativo
- Multiplicación matriz-matriz

Tarea 1. Multiplicación Matriz-vector

Desarrollar un programa que implemente el mismo esquema iterativo de la práctica 1, desarrollado para OpenMP, es decir para $k=0..m$:

$$\begin{aligned}x^{k+1} &= M x^k \\ x^{k+1} &= (\alpha * x^{k+1}) + ((1 - \alpha) * x^k)\end{aligned}$$

Los valores de M , que es una matriz cuadrada de tamaño $N \times N$ se generarán pseudoaleatoriamente y estarán comprendidos entre $0 < x < 1$. El vector x^0 será el vector unidad ($= 1, 1, \dots, 1$). Tras cada iteración se calculará la norma uno del vector diferencia ($|x^{k+1} - x^k|$, es decir el sumatorio de los valores absolutos de la resta de cada uno de las componentes) y se mostrará en pantalla. El número de iteraciones (m) y el parámetro α serán parámetros pasados en sentencia de ejecución.

Posteriormente deberemos analizar en términos de tiempo, speed-up y eficiencia. Para ello tomaremos un tamaño de matriz $N=14000$ y ejecutaremos de 1 a 12 procesos tomando $m=98$ y $\alpha = 0.99$. En la memoria se mostrarán las gráficas con las tablas de tiempo. El tiempo a medir no incluirá la generación de la matriz ni la reserva de memoria.

Se debe realizar la multiplicación matriz vector más suma de dos modos diferentes: a) cada proceso calculará un elemento del vector resultado y b) todos los procesos cooperarán para calcular cada uno de los elementos del vector resultado. (El cálculo de la norma no variará de un caso a otro)

Además ambas versiones habrá que realizarlas utilizando automatizadores paralelos y sin utilizarlos (es decir especificando explícitamente el trabajo que ha de realizar cada uno de los procesos).

Tarea 2. Multiplicación Matriz-Matriz

Basado en el programa anterior desarrollar un programa que implemente la multiplicación matriz-matriz sobre una arquitectura de memoria compartida utilizando OpenMP. Analizaremos en términos de tiempo, speed-up y eficiencia el algoritmo. Para ello tomaremos un tamaño de $N=1200$ y

ejecutaremos de 1 a 12 procesos. El tiempo a medir no incluirá la generación de la matriz ni la reserva de memoria.

Se debe realizar la multiplicación como en el caso anterior con dos opciones: a) cada proceso calculará un elemento de la matriz producto y b) todos los procesos cooperarán para calcular cada elemento de la matriz producto. Además, como en el caso anterior, ambas versiones habrá que realizarlo utilizando automatizadores y sin utilizarlos, es decir especificando explícitamente el trabajo que ha de realizar cada uno de los procesos.

En el caso de la opción a) se considerarán las siguientes opciones (con y sin automatizadores):

- Un proceso calcula un grupo de filas adyacentes
- Un proceso calcula grupos de filas adyacentes igual a 20
- Un proceso calcula un grupo de columnas adyacentes
- Un proceso calcula grupos de columnas adyacentes igual a 20

En una ejecución se calcula el producto usando todos los modos y muestra el tiempo de cada uno de ellos al final de la ejecución.

ATENCIÓN = Los datos de las matrices y vectores se generarán pseudoaleatoriamente y serán de tipo DOUBLE.

AUTOMATIZADORES PARALELOS: bucles for paralelos, reduction

NOTA: En caso de utilizar sulli en lugar de randall ejecutaremos de 1 a 5 procesos.