

## **Computación Paralela**

<b>Profesor Responsable:</b> Héctor Fco Migallón Gomis <a href="mailto:hmigallon@umh.es">hmigallon@umh.es</a>			
<b>Profesor de Laboratorio:</b> Héctor Fco Migallón Gomis <a href="mailto:hmigallon@umh.es">hmigallon@umh.es</a>			
<b>Departamento:</b> INGENIERÍA DE COMPUTADORES			
<b>Área de Conocimiento:</b> Arquitectura y Tecnología de Computadores			
<b>Curso:</b> 3º	<b>Docencia:</b> 1Sem.	<b>Tipo:</b> Obligatoria	<b>Créditos:</b> 6 ECTS (60 + 90 horas)
<b>Página web de la asignatura:</b> (institucional)			

### • **PRACTICA 1**

#### **Tarea 0. Procesado de imagen y Sistema iterativo secuencial**

Desarrollar la tarea 1 y la tarea 2 en secuencial.

#### **ENTREGA:**

Ficheros fuente de los códigos (COMENTADOS)

#### **Tarea 1. Procesado de imagen**

El objetivo de esta práctica es programar con el paradigma MPI para sistemas de memoria distribuida. Se leerá una imagen en tonos de grises almacenada en fichero binario con extensión raw, es decir los datos serán de tipo unsigned char y el tamaño del fichero será de altura x anchura x 1byte.

La imagen leída se almacenará en una matriz bidimensional por el proceso root, las imágenes procesadas se almacenarán en ficheros binarios en formato raw, proceso realizado por el proceso root.

Procesados a implementar (se escogerá uno pasando parámetro):

- Filtrado por media (3 x 3 elementos)
- Filtrado por mediana (3 x 3 elementos)
- Detección de bordes (SOBEL)

C

-1	0	1
-2	0	2
-1	0	1

F

-1	-2	-1
0	0	0
1	2	1

$$F_{ij} = \text{sqrt}(C^2 + F^2)$$

El fichero texto de salida, generado por el proceso root contendrá la siguiente información: nombre de fichero de entrada, tamaño de la imagen, nombre de fichero de salida, número de procesos utilizado, tiempo de ejecución (contado desde que el root ya tiene la imagen en memoria hasta antes de comenzar a guardar el fichero con la imagen procesada).

En la detección de bordes por SOBEL, aquellos elementos que no disponen de los 8 vecinos SI serán considerados, realizando extensión simétrica tanto para filas (la fila -1 será igual a la 1, la fila N será igual a la N-2, siendo N el número de filas) como para columnas.

Los nombres de los ficheros y resto de parámetros necesarios (tamaño de la imagen y tipo de procesado) serán pasados como argumentos en la sentencia de ejecución.

Como se ha dicho, el proceso “root” será el encargado de leer y escribir ficheros, el proceso “root” almacenará toda la información y distribuirá un conjunto de filas a cada proceso “slave”, recibirá los datos de cada proceso “slave” y escribirá en fichero.

#### ENTREGA:

Ficheros fuente del código (COMENTADOS)

Fichero que incluya los resultados de todas las opciones (media, mediana, sobel) para 5 procesos procesando la imagen lena\_4096x4096.raw, la sentencia de compilación, un ejemplo de sentencia de ejecución y una tabla de resultados con tiempos y valores de speed-up y eficiencia para las tres opciones y de 1 a 6 procesos (TODO JUNTO EN UN SOLO FICHERO TEXTO O PDF).

## Tarea 2. Sistema iterativo

Desarrollar un programa que implemente el siguiente esquema iterativo para  $k=0..m$ . Teniendo en cuenta que el vector  $x^0$  será el vector unidad ( $= 1,1,...,1$ ). El valor del número de iteraciones ( $m$ ) se especificará como argumento. El tamaño de los vectores será igual a  $N=15000$ .

Iteración 1:  $x^1 = M x^0$   
Iteración 2:  $x^2 = M x^1$ ; Buscar el mayor valor absoluto de  $x^2$ , si es superior al valor literal 25.0, convertir  $x^2$  a valores entre  $[-1,1]$ , dividiendo todos los elementos por dicho valor.  
Iteración 3:  $x^3 = M x^2$ ; Buscar el mayor valor absoluto de  $x^3$ , si es superior al valor literal 25.0, convertir  $x^3$  a valores entre  $[-1,1]$ , y dividiendo todos los elementos por dicho valor..  
...

La matriz  $M$  (cuadrada de tamaño  $N \times N$ ) estará almacenada en un fichero binario, si no existe dicho fichero se generan los valores de  $M$ , con elementos iguales a 1 en la diagonal y el resto se generarán aleatoriamente o pseudoaleatoriamente y estarán comprendidos entre  $-50.0 < x < 50.0$ , de forma que los elementos no diagonales de la mitad triangular inferior serán positivos y de la triangular superior negativos.

El número de iteraciones ( $m$ ) y nombres de ficheros serán pasados en sentencia de ejecución.  $M$  será una matriz reservada fila a fila, es decir puntero doble. Los elementos de la matriz y los vectores son de tipo double. Sólo el proceso root lee y escribe ficheros.

El fichero texto de salida, generado por el “root” tendrá la siguiente información: nombre de fichero de la matriz, nombre de fichero de salida, elementos de mayor valor absoluto obtenidos, número de procesos utilizado, tiempo de ejecución desde justo antes de iniciar iteraciones hasta el final de la ejecución.

Es obligatorio usar comunicaciones colectivas en aquellos procesos que lo admitan.

IMPORTANTE: Usar los recursos (memoria) estrictamente necesarios.

#### ENTREGA:

Ficheros fuente del código (COMENTADOS)

Fichero que incluya los resultados con  $m=8$  y usando 1 y 7 procesos, la sentencia de compilación, un ejemplo de sentencia de ejecución y una tabla de resultados con tiempos y valores de speed-up y eficiencia con  $m=8$  y de 1 a 6 procesos (TODO JUNTO EN UN SOLO FICHERO TEXTO O PDF).