ARQUITECTURA DE COMPUTADORES II

PRÁCTICA: Reconocimiento paralelo de patrones en OpenMP

AUTORES:

David Expósito Singh

Florin Isaila

Daniel Higuero Alonso-Mardones

Javier García Blas

Borja Bergua Guerra

Área de Arquitectura y Tecnología de Computadores

Departamento de Informática

Universidad Carlos III de Madrid

Julio de 2012

cc BY NC SA

Reconocimiento paralelo de patrones en OpenMP

Descripción

El objetivo de la práctica será desarrollar un programa paralelo (haciendo uso de openMP), que ejecute un algoritmo de reconocimiento de patrones. Para ello se aplicará una técnica de "fuerza bruta": se recorrerá cada pixel de la imagen en busca de las figuras. Para ello, se dividirá la imagen en distintos cuadrantes, y de este modo, repartir la carga de trabajo entre todos los hilos.

La solución del problema se plantea de la siguiente forma:

- La imagen se leerá en el hilo principal (función main)
- Todos las áreas se deben haber procesado completamente antes de escribir el fichero de centros en disco.
- La escritura de los centros a fichero se puede realizar tanto en el hilo principal como en una sección paralela (usando la directiva #pragma omp master).

Se deberá implementar tres versiones del programa (aunque a la hora de la entrega solo se haga de una) con diferentes estrategias de planificación de openMP:

1. Estática
2. Dinámica
3. Planificación asistida o guiada

El programa desarrollado tomará como argumentos los siguientes parámetros, y en el siguiente orden:

- El nombre de fichero que contiene la imagen.
- Un fichero de salida en el que se escribirá los objetos reconocidos mediante el formato (x y), donde x e y son las coordenadas del centro de la figura.
- El número de hilos.

De esta forma, la invocación al programa será:

./bmp_openmp <fichero_imagen_original> <fichero_centros_salida> <num_hilos>

Requisitos

Para aprobar la práctica es necesario cumplir los siguientes requisitos:

- Se deberá usar la primitiva gettimeofday para la obtención de tiempos.
- Se tomarán medidas de rendimiento sobre 1, 2, 4, y 8 hilos y 2 imágenes distintas: una con una distribución uniforme y otra con una distribución geométrica.
- El speedup se calculará de dos formas distintas (speedup teórico y speedup real)

    - Speedup teórico: Tiempo de cómputo / número de hilos
    - Speedup real: Tiempo de cómputo obtenido en la práctica
- El tiempo de acceso a los ficheros no se tendrá en cuenta a la hora de calcular el tiempo total de ejecución.

Entrega

El código será entregado mediante un único fichero llamado bmp_openmp.c.

La memoria tendrá que contener los siguientes apartados:

- Descripción de la metodología de paralelización.
- Gráficas con la medida de tiempos y mejora del rendimiento (speedup).
- Gráficas con el tiempo total de ejecución.
- Interpretación de las gráficas (muy importante).
- Comparación entre los paradigmas de programación de paso de mensajes, hilos y openMP (esta práctica).
- Conclusiones.

Pistas

1. Es muy importante que para el funcionamiento de la práctica se ejecute previamente el siguiente comando, dónde indicaremos a la máquina el número máximo de hilos disponibles:

```
export OMP_NUM_THREADS=16
```

2. En caso de compilar la aplicación en las aulas de los laboratorios es necesario usar el siguiente comando:

```
omcc -c bmp_openmp.c
```

3. En caso de compilar la aplicación en el cluster (kasukabe), para compilar usaremos el flag “-fopenmp”:

```
gcc -c -fopenmp bmp_openmp.c
```