

Computación paralela

Práctica 1

1. Implementación del efecto borroso de una imagen

Con esta práctica se va a comparar el rendimiento de un algoritmo para la generación del efecto borroso de una imagen, bajo las siguientes condiciones de paralelización:

1. CPU - secuencial.
2. CPU - Hilos POSIX (2, 4, 8 y 16 hilos).

Probar con imágenes en color de 720p, 1080p y 4K. Puede hacer uso de una librería para manipulación de imágenes. El algoritmo para efecto borroso debe elaborarse.

Se debe pasar como argumentos del programa el nombre de la imagen, el tamaño del kernel del efecto (entre 2 y 16) y el número de hilos. Por ejemplo:

```
cesar@desktop:~$time blur-effect miImagen.png 16 4
```

El comando anterior ejecuta la implementación de hilos en CPU para la imagen `miImagen.png` con un filtro con kernel de tamaño 16 y 4 hilos.

1.1. Script para ejecución.

Para la ejecución de los programas, elaborar un *script* que permita ejecutar uno a uno los casos y almacenar los resultados de tiempo de respuesta en un archivo de texto plano.

1.2. Consideraciones.

- Especificar la forma en que se hace el balanceo de carga, especificando el método con el que fue particionado el problema (block, block-wise). Hacer dibujos explicativos.
- Hacer uso de punteros y de memoria dinámica (`malloc()` - `free()` en *host*).
- Elaborar gráficas donde se observe:

- El tiempo de respuesta y el número de hilos para cada tamaño de imagen.
- El *speedup* alcanzado con el número de hilos para cada tamaño de imagen.
- Entregar los siguientes archivos:

```
blur-effect.c  
script_ejecutar_todo  
reporte.pdf  
LEEME
```

Archivo LEEME con observaciones, todos dentro de una carpeta con los nombres que aparecen en el correo para cada integrante. Pej la carpeta podría llamarse: **capedrazab-capedrazab**. Entregar el archivo tar con el mismo nombre de la carpeta.

1.3. Calificación.

Se tendrán en cuenta los siguientes aspectos para la evaluación:

- Funcionamiento pleno del programa (requisito para entrega) 40
- Script de ejecución para todos los programas almacenando el tiempo de respuesta para cada caso. 10 %
- Código limpio. 10 % (modular, tabulaciones, comentarios - básicos, declaración de constantes, buenos nombres para variables, etc.)
- Sustentación por parte de cualquiera de los integrantes del grupo de trabajo. 20
- Reporte. Archivo pdf con formato IEEE puesto en la página donde se hagan explicaciones a cerca del diseño (particionamiento, etc), experimentos, resultados (tiempos de respuesta, speedup) y conclusiones. 20 %