

Trabajo Final de Introducción a la Concurrency 2020

Profesor: Lic. Marcelo Eleazar Gómez
Auxiliar: Lic. Gastón Díaz

November 24, 2020

Pautas

Se podrán conformar grupos para resolver el trabajo final. Se entregará en formato digital en el Aula Virtual, tanto los programas desarrollados como el documento PDF explicativo del problema y su solución. Además se preparará una presentación con diapositivas y se expondrá el tema asignado al resto de la clase mediante alguna plataforma de videoconferencia como jitsi. Las diapositivas también formarán parte de la entrega.

Fecha de presentación: Viernes 11 de diciembre de 2020

Trabajos

1. Cálculo Científico: Jacobi Iteration

Se deberá desarrollar una presentación explicando en qué consiste y qué problema resuelve. Además deberá programar una solución secuencial y una paralela usando memoria compartida. Se puede usar pthreads u OpenMP. Se deberán realizar pruebas de rendimiento de las diferentes soluciones, con diferentes parámetros.

Bibliografía: Cap. 11 de Foundations of Multithreaded, Parallel and Distributed Programming, desde pág. 534 hasta 541

2. Cálculo Científico: El problema gravitacional de N cuerpos

Se deberá desarrollar una presentación explicando en qué consiste y cómo se resuelve. Además deberá programar una solución secuencial y una paralela usando memoria compartida. Se puede usar pthreads u OpenMP. Se deberán realizar pruebas de rendimiento de las diferentes soluciones, con diferentes parámetros.

Bibliografía: Cap. 11 de Foundations of Multithreaded, Parallel and Distributed Programming, desde pág. 553 hasta 559

3. Procesamiento de imágenes: Etiquetado de regiones

Se deberá desarrollar una presentación explicando en qué consiste. Además deberá programar dos soluciones paralelas usando MPI. El programa deberá correr en diferentes PCs comunicandose mediante pasaje de mensajes (MPI). Hacer una versión paralela con un proceso coordinador, y otra sin él. Se deberán realizar pruebas de rendimiento de las diferentes soluciones, con diferentes parámetros. Para simular una imagen se utilizará una matriz de enteros. Cada celda tendrá un cero (pixel apagado) o un uno (pixel encendido). Las matrices se asumirán cuadradas y el tamaño n será un parámetro para la ejecución del programa. Confeccionar un conjunto de imágenes de ejemplo y medir la performance de cada programa con las diferentes imágenes y con diferente cantidad de procesos.

Bibliografía: Cap. 09 de Foundations of Multithreaded, Parallel and Distributed Programming, desde pág. 430 hasta 435

4. **Procesamiento de imágenes: Suavizado (Smoothing)**

Se deberá desarrollar una presentación explicando en qué consiste. Además deberá programar dos soluciones paralelas usando MPI. El programa deberá correr en diferentes PCs comunicandose mediante pasaje de mensajes (MPI). Hacer una versión paralela con un proceso coordinador, y otra sin él. Se deberán realizar pruebas de rendimiento de las diferentes soluciones, con diferentes parámetros. Para simular una imagen se utilizará una matriz de enteros. Cada celda tendrá un cero (pixel apagado) o un uno (pixel encendido). Uno de los parámetros será la cantidad de vecinos (d) que determinará el apagado de un pixel. Las matrices se asumirán cuadradas y el tamaño n será un parámetro para la ejecución del programa. Confeccionar un conjunto de imágenes de ejemplo y medir la performance de cada programa con las diferentes imágenes, con diferentes valores de d , y con diferente cantidad de procesos.

Bibliografía: Cap. 09 de Foundations of Multithreaded, Parallel and Distributed Programming, desde pág. 430 hasta 435, y pág. 480