

Tarea 4

IMT2112 - Algoritmos Paralelos en Computación Científica

Elwin van 't Wout

27 de noviembre de 2019

Introducción

Hay una variedad de algoritmos para aproximar el valor del número π . Una familia de métodos numéricos es Monte Carlo, los cuales usan números aleatorios en sus aproximaciones. Un ejemplo tradicional es aproximar el número π por medio de generar números aleatorios en el plano y contar la cantidad de puntos adentro del círculo. Más precisa, genera pares de números aleatorios (x_i, y_i) para $i = 1, 2, \dots, N$ en la cuadra $[0, 1] \times [0, 1]$. El número de puntos que satisfacen $x_i^2 + y_i^2 < 1$ dividido por N es una aproximación del área de un cuarto círculo, la cual es $\pi/4$. Por lo tanto, multiplicar la proporción por cuatro resulta en una aproximación de π .

Actividades

Para la tarea, se tiene que aproximar π con el método de Monte Carlo en OpenCL. Dado que el Mazinger no tiene un nodo de GPU, pueden usar tu propio computador. Primero, corre el código `openc1-devices.cpp` disponible en Siding y copia el resultado al informe.

1. Genera los puntos (x_i, y_i) aleatoriamente.
Sugerencia: en C++ se puede generar números aleatorios con la biblioteca `cstdlib`. La función `rand()` genera un número aleatorio entre cero y el valor máximo dado por el constante global `RAND_MAX`.
2. Explica por qué no se puede paralelizar la generación de números aleatorios y, por lo tanto, no será eficiente hacerlo en la GPU.
Sugerencia: la función es un ‘pseudo random number generator’.
3. Programa el método de Monte Carlo en OpenCL.
 - a) El cálculo del radio $r_i = x_i^2 + y_i^2$ debe ser ejecutado en la GPU.
 - b) La condición $r_i < 1$ debe ser ejecutado en la GPU. Explica por qué esto podría ser ineficiente en la GPU.
 - c) Sumar los puntos adentro del círculo debe ser ejecutado en la GPU. Explica como has programado esta operación de reducción de forma eficiente.
4. Calcula el error de la aproximación. Explica como cambia el error a la hora de cambiar el número de puntos N .

Reglamento

La tarea debe cumplir con las reglas siguientes.

- Mientras se puede trabajar junto a compañeros en la preparación de las tareas, el código y el informe deben ser elaborados de forma individual.
- No es permitido copiar código de fuentes externos, tampoco sacar código de internet. El código entregado debe ser programado por tu mismo. En esta tarea se puede usar los códigos disponibles en Siding.
- Entrega un archivo comprimido con el código de C++ y un informe corto, elaborado en \LaTeX , con las respuestas a las preguntas. Suben el archivo al cuestionario en Siding y usa como nombre del archivo `Tarea4_apellido.zip` en lo cual se tiene que reemplazar `apellido` con tu propio apellido.
- El plazo de entrega es detallado en el cuestionario correspondiente a esta tarea en Siding. Es posible subir la tarea múltiples veces hasta el plazo del cuestionario.