Laboratorio 0: Hello World

Objetivo

Este ejercicio consiste en inicializar el entorno para asegurar que se pueden compilar aplicaciones OpenCL. Los códigos fuentes se encuentran en "igpuLab0". No se requiere entregar!!!.

1. Trabajo preliminar

Step 1. Instalar los paquetes necesarios para compilar código OpenCL

- Netbeans
- NVIDIA SDK / Intel OpenCL SDK
- OpenCL for JAVA (jogAMP package)

Step 2. Descargar el Laboratorio 0 e incluirlo en el entorno
Step 3. Run
Step 4. Verificar los resultados con una implementación secuencial

2. Primer programa OpenCL

Datos de entrada **:**

vector<float> A ;

Datos de salida :

vector<float> A;

Kernel :

A[i] = A[i] + 1;

Laboratorio 1: Operaciones de vectores

Objetivo

Este ejemplo de laboratorio consiste en realizar un conjunto de operaciones que tengan como entra 2 vectores, y almacenarlo en un tercer vector. Los códigos fuentes se encuentran de base se encuentran en la carpeta "igpuLab1".

Las operaciones que se deben implementar son, con los siguientes parámetros:

- *Vector-Vector-Vector* : Suma / Resta / Multiplicación elemento a elemento
- *Vector-Cardinal-Vector* : Suma / Multiplicación / División
- *Vector-Vector :* Invertir el orden del vector y dejarlo en el vector original

Implementación en OpenCL

1. Descargar el Laboratorio 1, el cual contiene el ejemplo de Suma de Vectores.

2. Modificar el ejemplo base para incluir nuevos tipos de operaciones.

3. Run

4. Verificar los resultados con una implementación secuencial

5. Extraer tiempos

Laboratorio 2: Bandwidth & Timing

Objetivo

Este ejemplo de laboratorio consiste en realizar un conjunto de pruebas donde se copian y leen datos de diverso tamaño, y se toman los tiempos de copia en cada caso. Además se analiza la influencia de los accesos a memoria en relación a las operaciones matemáticas.

Consideraciones

- Las copias deben ser sincrónicas.
- Verificar los accesos a los límites.
- Utilizar estructuras auxiliares en caso que sea necesario.

Las variaciones que deben hacerse son :

- Del tamaño de entrada de los datos. Por lo menos 3
- Del tamaño de los grupos (*local work size*). . Por lo menos 3 de 1,2,4,8,16,32 & 64
- De las operaciones por dato. Implementar 3 casos con diferentes cantidades de accesos a memoria y operaciones matemáticas. Por ejemplo:
    - XOut[index] = x[index] + 1
    - XOut [index] = pow(x[index], 5) * sin (x[index]) * sqrt(x[index])
    - XOut [index] = x[index -1] + x[index +1]

Implementación en OpenCL

1. Descargar el Laboratorio 2, el cual contiene el ejemplo.
2. Modificar el ejemplo variando la cantidad de datos transferidos.
3. Run
4. Tomar los tiempos en cada caso. Construir una tabla de datos con el tamaño de datos y el tiempo que requirió. Calcular el Bandwidth y comparar con el teórico.

Laboratorio 3: Filtrado de imágenes

Objetivo

Este ejemplo de laboratorio consiste en realizar un conjunto de operaciones sobre imágenes. Los códigos fuentes se encuentran de base se encuentran en la carpeta "igpuLab3". Utilizar una estructura auxiliar en caso que sea necesario.

Se deben implementar:

- Filtro de brillo<Image , float *brillo*>.
    - P[x,y] = *brillo + P[x,y]*
- Threshold <Image , float tolerance>
    - *Todo valor debajo de tolerance es 0 en otro caso 1.*
- Filtro de contraste<Image , float MinI , float MaxI) . MinI y MaxI son 2 valores cualquiera
    - P[x,y] = (P[x,y] - MinI) / (MaxI - MinI)
- Suavizado.
    - P[x,y] =(P[x-1,y] + P[x+1,y] + P[x,y-1] + P[x,y+1] + P[x,y]) /5

Implementación en OpenCL

1. Descargar el Laboratorio 3, el cual contiene el ejemplo de Imagen negativa.
2. Modificar el ejemplo base para incluir nuevos tipos de operaciones.
3. *Run*
4. Verificar visualmente
5. Extraer tiempo