ITESO primavera 2019

Primer Examen - Programación de procesadores gráficos.

Nombre: Carlos Alberto Cordero Robles

Sección 1: Conteste las siguientes preguntas o elija la opción correcta según corresponda.

**1.- ¿Por qué son necesarias las arquitecturas multi-core (CPU) y many-core (GPU) en el computador moderno? (Valor: 5 pts)**

En genera su propósito es distinto, esto debido a las características de cada uno de estos elementos. Tanto el CPU como el GPU ejecutan instrucciones, pero el poder de cada uno es distinto. Por lo General el CPU es mucho más rápido que el GPU, pero el GPU tiene muchos más núcleos que el CPU y todos ejecutaran la misma tarea lo que le da mayor poder de paralelismo. En cuestión de memoria, el CPU es eficiente en el manejo de datos pequeños y no necesariamente juntos, en cambio el GPU espera procesar grandes bloques de memoria.

Sabiendo estas diferencias entonces se puede definir que actividades es mejor tome el CPU y cuales es mejor que tome el GPU.

El CPU es muy necesario para el procesamiento del sistema operativo, ya que este asignara tiempos a cada tarea. Por lo general ni son la misma tarea ni involucran paralelismo a nivel tarea, por lo que seria muy poco práctico hacer lo con el GPU. En general toda la administración de las tareas multimedia será manejada por el CPU.

Por otra parte, hay tareas que requieren de un gran paralelismo para poder tener resultados de calidad. El manejo de gráficos es un ejemplo de esto. Cada píxel de la pantalla es independiente necesita ser procesado, dejarle esto al CPU podría afectar la calidad de la imagen, en casos como estos es cuando se vuelve indispensable un GPU.

EN resumen, en la actualidad un computador moderno requiere del manejo de SO y programas multimedia pero así mismo necesita el poder de un GPU para procesar grandes porciones de datos de manera paralela es por esto por lo que tanto arquitecturas de CPU como de GPU son necesarias en un computador moderno.

**2.- ¿En lo general, un GPU puede hacer lo mismo que un CPU? Justifique. (Valor: 5 pts)**

En lo personal pienso que no. La arquitectura y las microinstrucciones de un CPU serán mas complejas que las del GPU, seguramente habrá instrucciones que el GPU no podrá realizar. Aparte las tareas para las que están diseñado el CPU involucran acceso a datos distribuidos y administración de tareas, esto sería prácticamente imposible para el GPU ya que los hilos ejecutan la misma tarea y por lo general trabajan sobre un bloque de datos.

**3.- ¿En lo general, un CPU puede hacer lo mismo que un GPU? Justifique. (Valor: 5 pts)**

En lo personal pienso que sí, pero mucho mas lento, si esto no afecta el resultado esperado entonces en ese caso en particular si. Aunque los CPUs son por lo general más rápidos que el GPU, la cantidad de núcleos es mucho menor por lo que si se necesita manejar grandes cantidades de datos lo hará y de manera paralela también como el GPU, pero la eficiencia será mucho menor.

**4.- Si el rendimiento de los procesadores gráficos (GPU) es superior a los procesadores convencionales (CPU), ¿Por qué las computadoras modernas siguen necesitando a los CPU? (valor 5 pts)**

Como se menciona en la pregunta 1. la mejora de rendimiento del GPU es mayor que la del CPU debido a su paralelismo a nivel tarea. Pero habrá tareas que no podrán ser manejadas por el GPU por su misma naturaleza de acceso a la memoria y el manejo del SO junto con sus múltiples tareas distintas que administra, esto solo podrá manejarlo el CPU.

5.- Seleccione el tipo de procesador más adecuado a utilizar en las siguientes aplicaciones indicando CPU o GPU según sea el caso. Justifique su elección en cada caso. (Valor: 30 pts, 2 pt por elección correcta y justificación)

a) Procesamiento asincrónico de tareas o actividades diversificadas ( CPU )

El CPU junto con el CPU puede manejar las tareas y actividades el GPU se especializa en paralelizarlas.

b) Control de conexiones de red en el computador ( CPU )

La conexión seguramente se lleva a cabo por un Handshake que depende de interrupciones y el manejo del canal de comunicaciones no depende de paralelismo por lo que un GPU no es necesrio.

c) Servidor WEB (HTTP) ( CPU )

Los servidores tendrán que atender varias peticiones asíncronas y no siempre será la misma tarea a ejecutar.

d) Ejecución de un sistema operativo multitarea con prioridad ( CPU )

El manejo del sistema operativo esta hecho por el CPU ya que el sistema operativo se encargara de distribuir los tiempos para las distintas actividades, lo cual es mas sencillo para un CPU.

e) Ordenación de datos ( GPU )

Las mismas instrucciones SIMD/MIMD que maneja el GPU le permite ordenar la información de manera mas rápida, ya que se ordena el dato por HW y esto lo hace de manera muy eficiente.

f) Búsqueda de datos en un conjunto ordenado ( GPU )

La búsqueda es la misma tarea pero aplicada a cada dato, esto lo puede hacer de forma muy eficiente el GPU.

g) Operaciones estadísticas sobre conjuntos de datos ( GPU )

Definitivamente un GPU, los datos a estudiar no dependen uno de otro y las operaciones se

Serán prácticamente sumas, multiplicaciones y divisiones por lo que se puede paralelizar fácilmente.

h) Virtualización de computadoras ( CPU )

La visualización de la computadora está muy relacionada con el SO, por lo que estará mas relacionado con el CPU.

i) Administración y operaciones sobre sistemas de archivos ( CPU )

El manejo de memoria del CPU es mas flexible, en el GPU es por grandes bloques de memoria. En este caso los sistemas de archivos no necesariamente están todos en el mismo bloque, es por esto por lo que sería una tarea más sencilla para el CPU.

j) Codificación / Decodificación de video ( GPU )

El video es un bloque inmenso de información, no depende ningún pixel de otro ni ninguna imagen de otra, esto permite paralelizar las tareas y sería una tarea más fácil y eficiente para el GPU.

k) Procesamiento de imágenes ( GPU )

La imagen es un bloque inmenso de información, no depende ningún píxel de otro, esto permite paralelizar las tareas y sería una tarea más fácil y eficiente para el GPU.

l) Edición de texto ( CPU,GPU )

La edición de texto involucra varias tareas distintas corriendo en paralelo, por mencionar algunas: Lectura de teclado y ratón, dar formato y fuente al texto, guárdalo y carga de archivos, estas tareas son manejadas por el CPU. Pero a fin y al cabo es necesario imprimir en pantalla el texto, por lo que el GPU no queda de lado.

m) Codificación/Compresión de datos de audio ( GPU )

Normalmente los algoritmos de compresión de datos lo que hacen es analizar toda la trama de datos del audio y agrupar los 1 y los 0 (ZIP) las operaciones son sumas y multiplicaciones y no dependen una de otra por lo que se puede paralelizar y hacer más rápido con un GPU.

n) Control de transacciones bancarias o financieras ( CPU,GPU )

No conozco muy a fondo que esté detrás de una transacción bancaria pero seguro involucra mucha transición de información y validación de datos, estas distintas tareas tendrás que ser manejadas por un SO que usa recursos del CPU. Pero un punto muy importante es la ciberseguridad donde muchas veces una semilla tiene que ser procesada en un espacio de tiempo muy corto de tiempo, estos algoritmos de seguridad por lo general multiplican la llave por vectores de números primos y lo suman varias veces, ya cada sistema depende su algoritmo para generar la llave, este procesamiento para la llave puede ser manejado por un GPU para lograr mandar el dato en la ventana de tiempo deseada.

o) Compresión/Descompresión sin pérdida de datos de archivos ( GPU )

Normalmente los algoritmos de compresión de datos lo que hacen es analizar toda la trama de datos del audio y agrupar los 1 y los 0 (ZIP) las operaciones son sumas y multiplicaciones y no dependen una de otra por lo que se puede paralelizar y hacer más rápido con un GPU.

No se olvide de justificar brevemente cada una de sus elecciones.

Sección II Resuelva los siguientes problemas para GPU y construya el programa para demostrarlo en cada caso usando DirectCompute Shader Model 5.0. Recolecte evidencia y anexe archivos HLSL comentado y automatización de DirectCompute del lado del CPU (fragmento código C++ donde se prepara y realiza el despacho) e imágenes o instantáneas si aplica. Sea claro y conciso.

1. Construya un procesador de imágenes que reciba cualquier archivo BMP solicitado por el usuario y reduzca el color a escala de grises. La imagen procesada en el GPU debe presentarse en pantalla en escala de grises y no es necesario almacenar a disco los cambios.

Donde R, G, B son los canales de color de cada pixel.

(Valor: 30 pts)

El proyecto se llama “Grises”. Lo único que el usuario tiene que hacer es subir una imagen en formato BMP y poner la dirección y nombre del archivo en la variable “imageFile”:

Ejemplo:

const char imageFile[] = {"..\\Image.bmp"};

La imagen cargada se proyectará en escala de grises.

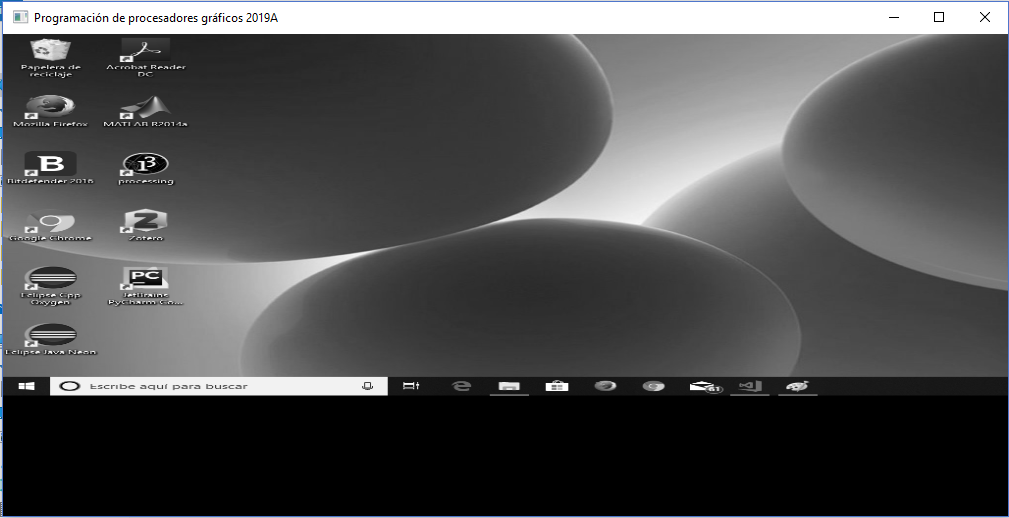


Imagen1. Salida del programa Grises

1. Construya un programa que procese todo un archivo de texto grande en codificación ANSI y que convierta el contenido de minúsculas a mayúsculas. La conversión debe realizarse en GPU. El programa almacena el archivo convertido con otro nombre de archivo.

(Valor: 30 pts)

El proyecto se llama “Mayusculas”. Lo único que el usuario tiene que hacer es subir un archio de texto formato ANSI y poner la dirección y nombre del archivo en la variable “pszFileName”:

Ejemplo:

const char pszFileName [] = { "..\\texto.txt" };

Después de cerrar la ventana desplegada se generará un archivo de texto a nivel del proyecto con el mismo texto pasado al proyecto, pero las minúsculas serán pasadas a mayúsculas. El archivo se llamará “mayusculas.txt”

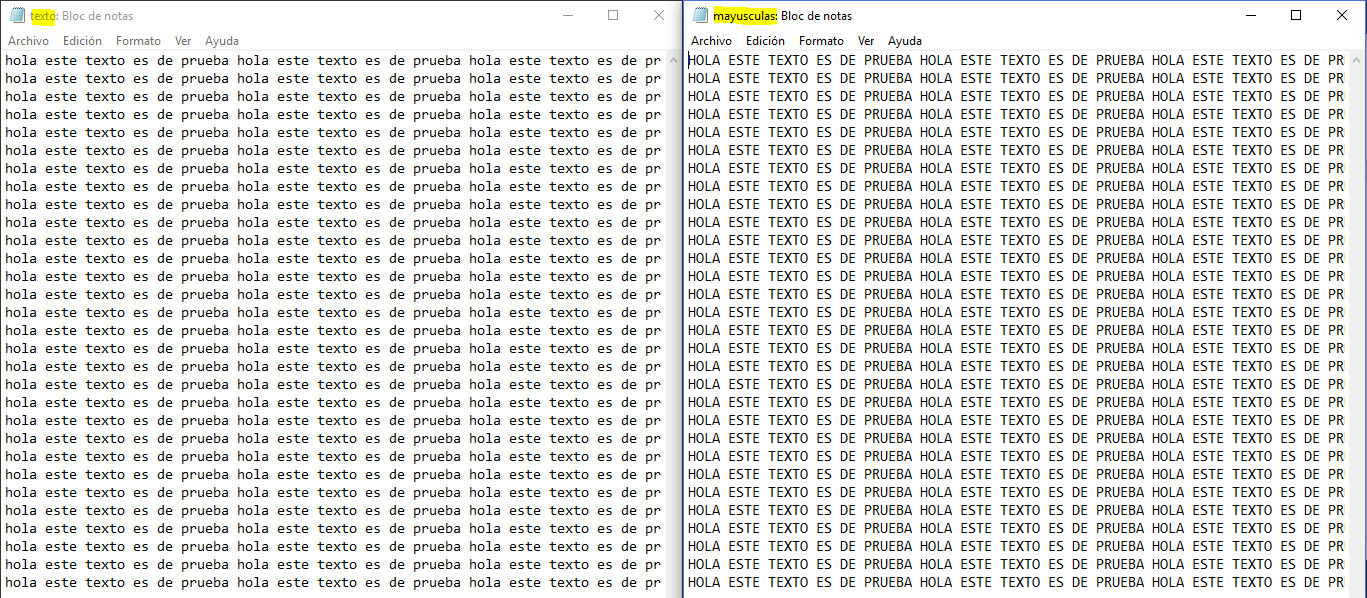


Imagen2. Comparación de documento de entrada(texto.txt) a la izquierda, contra documento de salida(mayúsculas.txt) a la derecha.

Para la entrega, suba su archivo zip, con los códigos fuente producidos para mayor comodidad junto a este documento con su nombre en el mismo zip. Por favor, utilice sólo formato zip en su entregable.