

Instituto Tecnológico de Costa Rica

Área Académica de Ingeniería en Computadores

(Computer Engineering Academic Area)

**Programa de Licenciatura en Ingeniería en
Computadores**

(Licentiate Degree Program in Computer Engineering)

Curso: CE-4302 Arquitectura de Computadores II

(Course: CE-4302 Computer Architecture II)



Evaluación Taller 4: CUDA

(Workshop 4 evaluation: CUDA)

Profesor:

(Professor)

Ing. M.Sc. Jeferson González Gómez

Fecha: 10 de octubre de 2018

(Date)

1. Micro-investigación

Previo a la realización de los ejercicios prácticos. Realice una pequeña búsqueda en internet, que responda las siguientes interrogantes:

1. ¿Qué es CUDA?
2. ¿Qué es un kernel en CUDA y cómo se define?
3. ¿Cómo se maneja el trabajo a procesar en CUDA? ¿Cómo se asignan los hilos y bloques?
4. Investigue sobre la plataforma Jetson TX2 ¿cómo está compuesta la arquitectura de la plataforma a nivel de hardware?

2. Conexión con plataformas

Para la sección práctica de este taller se trabajará con una plataforma embebida Nvidia Jetson TX2, a través de una conexión remota por protocolo ssh (secure shell). Como primer paso deberá instalar ssh en su estación de trabajo:

```
sudo apt-get install ssh
```

Una vez instalado, procederá a realizar la conexión con la plataforma. Hay disponibilidad para cuatro posibles plataformas:

- IP: 172.21.225.51
- IP: 172.21.225.52
- IP: 172.21.225.53
- IP: 172.21.225.54

Deberá seleccionar una de las plataformas y realizar la conexión ssh. Para esto deberá ejecutar:

```
ssh siplab@IP
```

donde IP corresponde a la dirección de la plataforma seleccionada. Para todas las plataformas, tanto el usuario como la contraseña es *siplab*

3. Hola mundo en CUDA

3.1. Instalación de aplicación ejemplo

Antes de iniciar con el ejemplo de CUDA, cada estudiante deberá crear un directorio propio dentro /home/siplab/, esto para no intervenir con los demás estudiantes. Esto se realiza:

```
mkdir juan
```

Luego, dentro del directorio propio, se deberá descargar el código de ejemplo que se utilizará para el resto del taller. Esto se realizará por medio de un repositorio de git:

```
git clone https://github.com/jefg89/helloCUDA
```

3.2. Ejemplo CUDA

Como primera prueba de CUDA, se va a trabajar con el archivo *vecadd.cu*, que utiliza un *kernel* en GPU para una operación de suma. Antes de continuar, debe asegurarse de entender correctamente qué es un kernel y cómo funcionan en el ámbito de CUDA.

Con base en el código de ejemplo:

1. Analice el código *vecadd.cu* y el *Makefile* correspondiente. A partir del análisis del código, extraiga cuáles son los pasos generales para la generación de aplicaciones utilizando CUDA.
2. Analice el código fuente del kernel *vecadd.cu*. A partir del análisis del código, determine ¿Qué operación se realiza con los vectores de entrada? ¿Cómo se identifica cada elemento a ser procesado en paralelo y de qué forma se realiza el procesamiento paralelo?
3. Realice la compilación del código *vecadd.cu*, utilizando el comando *make*
4. Realice la ejecución de la aplicación *vecadd*. ¿Qué hace finalmente la aplicación?
5. Cambie la cantidad de hilos por bloque y el tamaño del vector. Compare el desempeño antes al menos 5 casos diferentes.

4. Ejercicios prácticos

1. Realice un programa que calcule el resultado de la multiplicación de dos matrices de 4x4, utilizando paralelismo con CUDA.
2. Proponga una aplicación que involucre procesamiento vectorial. Implemente dicha aplicación tanto serial (sin paralelismo) como con CUDA. Mida tiempos de ejecución para diferentes tamaños y/o iteraciones.

Entregables

- En TecDigital: Documento de texto (pdf, doc, odf) que contenga las respuestas a la microinvestigación, preguntas teóricas del taller y resultados, así como link a repositorios de código fuente o captura de pantalla de los mismos.