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 5: OpenCL

(Workshop 5 evaluation: OpenCL)

Profesor:

(Professor)

Ing. M.Sc. Jeferson González Gómez

Fecha: 11 de mayo de 2018

(Date)

Micro-investigación

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

1. ¿Qué es OpenCL?
2. ¿Qué es un kernel en OpenCL y cómo se define?
3. ¿Cómo se diferencia entre asignación de trabajo a CPU y asignación a GPU?

Instalación

Como primer paso en este taller, se realizará la instalación de OpenCL, usando el API de AMD (brinda compatibilidad para procesadores AMD e Intel). En este taller se omitirá la instalación de drivers para el GPU, por lo que, de no contarlos, se deberá trabajar solamente utilizando el CPU (ver punto 3 de micro-investigación).

Para realizar la instalación, deberá descargarse el API de AMD, desde: `http://developer.amd.com/amd-accelerated-parallel-processing-app-sdk/`

Una vez descargado, extraer el archivo comprimido (para 64 bits):

```
tar xvf AMD-APP-SDKInstaller-v3.0.130.136-GA-linux64
```

Luego, ejecutuar el script (para 64 bits):

```
sudo ./AMD-APP-SDK-v3.0.130.136-GA-linux64.sh
```

Debe aceptarse los términos y licencia. **Para efectos de este taller el directorio de instalación será /opt/**, por lo que deberá especificarse a este paso.

NOTA: Si selecciona otro directorio de instalación, el defecto por ejemplo, deberá ajustar las instrucciones de compilación (que se describen más adelante) al directorio adecuado.

Verificación de la instalación

Para verificar que la instalación se haya dado adecuadamente, se utilizará el código fuente *devices.c* . Esta aplicación, una vez compilada, debería listar los dispositivos que su computador, que son compatibles con OpenCL.

Para realizar la compilación, de deberá ejecutar el comando (en una sola línea):

```
gcc -o devices devices.c -I/opt/AMDAPPSDK-3.0/include/
-L/opt/AMDAPPSDK-3.0/lib/x86_64/sdk -lamdocl64
```

NOTA: este comando toma en cuenta que instaló la versión de 64 bits del API de AMD en el directorio /opt/. En caso de tener la versión de 32 bits, o instaló en algún otro directorio, deberá adecuarse los directorios, y el enlace a la biblioteca dinámica (-lamdocl64), para tal fin.

Antes de realizar la ejecución, deberá incluirse la ruta donde se encuentra la biblioteca dinámica a la variable de ambiente de Linux correspondiente, esto se realiza con el siguiente comando:

```
export LD_LIBRARY_PATH=/opt/AMDAPPSDK-3.0/lib/x86_64/sdk
```

Ahora al ejecutar la aplicación *devices*

```
./devices
```

deberá mostrarse una salida similar a:

```
1. Platform
  Profile: FULL_PROFILE
  Version: OpenCL 2.0 AMD-APP (1800.8)
  Name: AMD Accelerated Parallel Processing
  Vendor: Advanced Micro Devices, Inc.
  Extensions: cl_khr_icd cl_amd_event_callback cl_amd_offline_devices
1. Device: Intel(R) Core(TM) i5-3340M CPU @ 2.70GHz
 1.1 Hardware version: OpenCL 1.2 AMD-APP (1800.8)
 1.2 Software version: 1800.8 (sse2,avx)
 1.3 OpenCL C version: OpenCL C 1.2
 1.4 Parallel compute units: 4
```

Hola mundo en OpenCL

Como primera prueba de OpenCL, se va a trabajar con el archivo *hello.c*, que utiliza un *kernel*, con el archivo correspondiente *hello.cl*. Antes de continuar, debe asegurarse de entender correctamente qué es un kernel y cómo funcionan en el ámbito de OpenCL.

1. Analice el código hello.c. A partir del análisis del código, extraiga cuáles son los pasos generales para la generación de aplicaciones utilizando OpenCl. Además, determine, ¿qué debería cambiar del código, para que el kernel sea ejecutado en un GPU en lugar de un CPU?

2. Analice el código fuente del kernel hello.cl. 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 hello.c, de la misma forma que se realizó anteriormente con el archivo devices.c

4. Realice la ejecución de la aplicación hello. ¿Qué hace finalmente la aplicación?

5. Repita los pasos anteriores, pero cambie el código para que se ejecute sobre GPU (sin importar si se cuenta con los drivers o no).

Ejercicios prácticos

1. Realice un programa que aplique la operación SAXPY tanto serial (normal) como paralelo (OpenCL), para al menos tres tamaños diferentes de vectores. Mida y compare el tiempo de ejecución entre ambos.

2. Realice un programa que calcule el resultado de la multiplicación de dos matrices de 4x4, utilizando paralelismo con OpenCL.

Entregables

- En TecDigital: Archivo con nombre Taller5_Nombre_completo.tar.gz que contenga los archivos de código fuente, Readme con comandos para compilación,y documento escrito con las respuestas de micro-investigación y otras preguntas teóricas de este taller.