

Propuesta de taller “Introducción a la Programación paralela de GPU’s con CUDA y OpenCL”

Saúl Calderón Ramírez

Investigador en el Centro de Investigación en Tecnologías de la Información y la Comunicación (CITIC), Universidad de Costa Rica (UCR)
saul.calderonramirez@ucr.ac.cr

1. Antecedentes y Justificación

Cada día es más común la necesidad de procesar grandes cantidades de datos en menor tiempo en muchas de las áreas en la ciencia. Tomando en cuenta lo anterior, la computación de alto rendimiento se puede definir entonces como la disciplina que suple de soluciones tanto en infraestructura como en entornos de programación a los científicos de las múltiples disciplinas con tales necesidades. Observando a grandes rasgos el desarrollo de la computación de alto rendimiento, puede decirse que sigue los pasos de la computación de consumo masivo de escritorio, la cual, a sus inicios contaba con un puñado de fabricantes que desarrollaban software y soluciones compatibles solo con sus plataformas, siendo incompatibles con el resto de plataformas existentes el mercado. El programador debía desarrollar soluciones para cada plataforma específica, incrementando considerablemente el trabajo necesario. Hoy en la computación de escritorio existen múltiples entornos de desarrollo x86, compiladores multi-plataforma, y máquinas virtuales como Java, las cuales permiten desarrollar soluciones multi-plataforma. La computación de alto rendimiento, basada actualmente en el paradigma de programación paralela con múltiples unidades de procesamiento como infraestructura, ofrece a los desarrolladores múltiples plataformas y dispositivos capaces de soportar procesamiento paralelo, como por ejemplo procesadores multinúcleo, servidores multi-procesador, procesadores gráficos (GPU's) y procesadores de uso específico (DSP's, PPU's, etc). Dada la variada oferta de plataformas, han surgido estándares de programación paralela como MPI (Message passing protocol), OpenMP, entre otros. Sin embargo, el auge de procesadores auxiliares (GPU's, DSP's, PPU's) ha provocado un uso frecuente de plataformas heterogéneas, con nodos de procesamiento compuestos múltiples procesadores auxiliares y varios CPU multinúcleo. OpenCL (Open Computing Language), es un marco de trabajo que permite desarrollar soluciones para múltiples plataformas, y también plataformas

heterogéneas. Su importancia gana terreno en la computación científica de alto rendimiento, pues OpenCL permite explotar de una manera más simple los recursos de infraestructura disponibles. El presente taller propuesto pretende introducir a académicos y estudiantes en la programación de OpenCL, en primer instancia introduciendo conceptos de la programación de GPU's utilizando CUDA, en los cuales se basa el modelo de plataforma de OpenCL, para luego introducir la estructura general de OpenCL y las funciones básicas necesarias para construir un programa en este marco de trabajo. El taller propuesto mostrará los detalles técnicos básicos para desarrollar una aplicación en OpenCL, pero dada la variedad de plataformas y procedimientos dependientes de ello para instalar OpenCL, no se requerirá del público desarrollar el ejemplo expuesto en este taller. Sin embargo estará a disposición el material expuesto en el taller, incluido el código fuente de los ejemplos, para el uso del público del taller. El taller propuesto ya fue expuesto en el PRIS-Seminar, el 14 de Junio del presente año en el mini auditorio de la escuela de Ingeniería Eléctrica, de la Universidad de Costa Rica, por Saúl Calderón. La presentación se basa básicamente en el libro Heterogeneous Computing with OpenCL (Benedict R. Gaster, 2012)

2. Objetivos General y Específicos

Objetivo general:

Presentar a la audiencia la motivación en el desarrollo del marco de trabajo OpenCL, su modelo de plataforma y las funciones básicas necesarias para construir una aplicación paralelizada en OpenCL.

Objetivos específicos:

1. Definir la motivación detrás de OpenCL: Explorar las distintas plataformas en infraestructura comunes para el procesamiento paralelo (GPU's, CPU's), sus diferencias básicas en cuanto a arquitectura además de las características básicas del entorno de programación CUDA, como preludio a la definición de la necesidad de un marco de trabajo como OpenCL.

2. Explorar el modelo de plataforma y ejecución de OpenCL: Explorar y discutir las distintas características del modelo de plataforma (entorno de programación) y del modelo de ejecución (entorno de ejecución de OpenCL). Explicar a grandes rasgos las funciones básicas para construir una aplicación en el marco de trabajo OpenCL.

3. Discutir las conclusiones de la presentación y las oportunidades con OpenCL: Exponer algunos consejos generales para la programación en OpenCL y comparar su rendimiento con otras plataformas similares como CUDA. Exponer y discutir las tendencias en la programación de alto rendimiento.

3. Audiencia esperada

La presentación desarrolla algunos conceptos básicos en la computación paralela para luego exponer el marco de trabajo OpenCL, sin embargo aunque poca experiencia en programación paralela es necesaria, si es recomendable que el público maneje conceptos básicos en programación. Es por esto que la audiencia objetivo se puede definir como estudiantes, académicos y profesionales de Computación, Ingeniería eléctrica y áreas afines, además de que científicos de otras áreas con experiencia en computación paralela pueden obtener provecho de la charla.

4. Cupo máximo

Dado que no es necesario disponer del equipo para desarrollar ejemplos de programación, el cupo máximo puede estar definido por la capacidad física del salón en el que se desarrollará la charla..

5. Calendario propuesto

El taller tiene como objetivo desarrollarse en 2 horas, tiempo en el que se realizó en el PRIS-Seminar. Según los objetivos específicos enumerados anteriormente, el desglose de tiempo para cada uno se enlista a en la Tabla 1: Calendario Propuesto.

Sección	Temas	Tiempo/minutos
1	Motivación detrás de OpenCL, introducción a CUDA y plataformas heterogéneas.	15
2	Modelo de plataforma de OpenCL y comparación con CUDA.	20
3	Modelo de ejecución de OpenCL.	15
4	Modelo de memoria y programación en OpenCL.	50
5	Consejos generales para la programación de OpenCL, comparación con otras tecnologías y discusión de las tendencias en la programación paralela.	20

Tabla 1: Calendario Propuesto.

6. Requerimientos técnicos y audiovisuales

No es necesaria la ejecución de ejemplos para este taller, por lo que como requisitos es necesario contar con proyector y un computador para ejecutar la presentación. Será opcional equipo con OpenCL instalado, sin embargo es prescindible.

Bibliografía

Benedict R. Gaster, L. H. (2012). *Heterogeneous Computing with OpenCL*. Morgan Kauffman.