Formato Aprobado por resolución Nº113 del CFI de fecha 04.07.2017

Programa de
Programación masivamente paralela en procesadores gráficos (GPUs)

1. NOMBRE DE LA UNIDAD CURRICULAR

Programación masivamente paralela en procesadores gráficos (GPUs)

2. CRÉDITOS

10 créditos

3. OBJETIVOS DE LA UNIDAD CURRICULAR

El objetivo de la unidad curricular es introducir al estudiante en el uso de los procesadores gráficos y otros dispositivos de hardware secundario para la resolución de problemas de propósito general.

Al finalizar el curso se espera que el estudiante:

- Pueda resolver problemas generales de complejidad media en unidades de procesamiento gráfico (GPU, por sus siglas en inglés).
- Sea capaz de implementar rutinas en el lenguaje CUDA C que hagan un buen aprovechamiento de los recursos de cómputo de las GPUs.
- Logre estudiar el tiempo de ejecución de rutinas que ejecutan en GPU, analizando sus cuellos de botella, y sea capaz de plantear optimizaciones con el fin de solucionarlos.
- Tenga un conocimiento general del ecosistema CUDA incluyendo librerías, profilers, debuggers, etc.
- Tenga un conocimiento general de otros dispositivos de hardware secundario que se utilizan para realizar cómputo de propósito general.
- Tenga un conocimiento general de conceptos de Computación Heterogénea.

4. METODOLOGÍA DE ENSEÑANZA

El curso dura 15 semanas. Durante estas se dictarán 12 clases de teórico de 2hs y se espera que los estudiantes destinen otras 24 horas al estudio de los temas presentados en las clases (total 48hs). Se espera que la resolución de los prácticos por parte de los estudiantes y la elaboración de los informes requiera un total de 65hs, incluyendo la asistencia opcional a 5 clases de presentación de práctico de 1 hora y 5 clases de consulta de 1 hora. Se estima que la elaboración del proyecto final de laboratorio insuma 40 hs.

5. TEMARIO

1. Introducción: Motivación. Repaso histórico del surgimiento de las GPUs, su uso para cómputo general y eficiencia.
2. Introducción al paralelismo:

a. Computación paralela: Paralelismo en máquinas secuenciales. Paralelismo en máquinas distribuidas. Paralelismo en máquinas distribuidas. Modelos y Estrategias para Programación Paralela. Paralelismo de Memoria Compartida. Paralelismo de Memoria Distribuida.

b. Programación paralela: Estrategias de scheduling y balance de carga. Distribución de datos/cálculos. Reducción. Condiciones de carrera. Operaciones atómicas.

3. Introducción a CUDA: Arquitectura CUDA. Modelo de ejecución.
4. Programación en CUDA:

a. Introducción: Device. Jerarquía de hilos (grids, bloques, warps). Uso de índices. Tipos de funciones. Keywords básicas y API.

b. Tipos de memoria: Acceso coalesced. Acceso a memoria (conflictos de bancos, tiling). Errores en tiempo de ejecución. Código PTX.

c. Transferencia de memoria: Transferencias sincrónicas y asincrónicas. Uso de memoria compartida.

5. Patrones de cómputo: Histograma. Reduce. Stencil. Scan.
6. Alternativas de otros fabricantes: Arquitecturas. Modelos de plataforma. Modelos de ejecución. Modelos de memoria. Modelos de programación.
7. Ecosistema CUDA: Herramientas (Debugging, Profiling), Librerías específicas (cusparse, curand, etc.) y Librerías de programación de alto nivel (thrust, cusp, modern gpu).
8. Nuevas Tendencias: Arquitecturas recientes. Paralelismo dinámico. Computación Heterogénea.
9. Aplicaciones de GPGPU: Álgebra lineal numérica. Computación gráfica. Deep Learning.

6. BIBLIOGRAFÍA

Tema	Básica	Complementaria
1. Introducción	-	-
2a. Computación paralela	(2)	-
2b. Programación paralela	(3)	-
3. Introducción a CUDA	(1)	(4, 5)
4a. Programación en CUDA (Introducción)	(1)	(4, 5, 6)
4b. Programación en CUDA (Tipos de memoria)	(1)	(4, 5, 6)
4c. Programación en CUDA (Transferencia de memoria)	(1)	(4, 5, 6)
5. Patrones de cómputo	(1)	(6)
6. Alternativas de otros fabricantes	-	-
7. Ecosistema CUDA	-	-
8. Nuevas Tendencias	(1)	(4, 5)

9. Aplicaciones de GPGPU	-	-

6.1 Básica

1. Hwu, Wen-mei W., Kirk, David B., El Haj, Izzat (2023). Programming Massively Parallel Processors: A Hands-on Approach Fourth Edition. United States: Morgan Kaufmann.
2. Hennessy, John L. and Patterson, David A. (2017). Computer Architecture A Quantitative Approach. 6th Edition. Elsevier.
3. Pacheco, Peter and Malensek, Matthew (2021). An Introduction to Parallel Programming. 2nd Edition. Elsevier.

6.2 Complementaria

4. NVIDIA (2024). CUDA C++ Programming Guide 12.3. Disponible en línea: https://docs.nvidia.com/cuda/cuda-c-programming-guide/
5. NVIDIA (2024). CUDA C++ Best Practices Guide 12.3. Disponible en línea: https://docs.nvidia.com/cuda/cuda-c-best-practices-guide
6. Sanders, J., & Kandrot, E. (2010). CUDA by example: an introduction to general-purpose GPU programming. Addison-Wesley Professional.

7. CONOCIMIENTOS PREVIOS EXIGIDOS Y RECOMENDADOS

7.1 Conocimientos Previos Exigidos: Conocimientos de programación en C y arquitectura de sistemas.

7.2 Conocimientos Previos Recomendados: Es recomendable el manejo de conceptos de sistemas operativos, computación gráfica y computación de alto desempeño (HPC).

Formato Aprobado por resolución Nº113 del CFI de fecha 04.07.2017

ANEXO A
Para todas las Carreras

Esta primera parte del anexo incluye aspectos complementarios que son generales de la unidad curricular.

A1) INSTITUTO

Instituto de Computación (InCo).

A2) CRONOGRAMA TENTATIVO

Consiste en un cronograma de avance semanal con detalle de las horas de clase asignadas a cada tema.

Semana 1	Tema 1 (2 hs de clase). Tema 2a (2 hs de clase).
Semana 2	Tema 2b (2 hs de clase). Práctico 1 (1 h de clase opcional).
Semana 3	Tema 3 (2 hs de clase). Consulta Práctico 1 (1 h de clase opcional).
Semana 4	Tema 4a y 4b (2 hs de clase). Práctico 2 (1 h de clase opcional).
Semana 5	Tema 4b (2 hs de clase). Consulta Práctico 2 (1 h de clase opcional).
Semana 6	Tema 4c y 5 (2 hs de clase). Práctico 3 (1 h de clase opcional).
Semana 7	Tema 5 y 6 (2 hs de clase). Consulta Pr. 3 (1 h de clase opcional).
Semana 8	Tema 7 (2 hs de clase). Práctico 4 (1 h de clase opcional).
Semana 9	Tema 7 (2 hs de clase). Consulta Práctico 4 (1 h de clase opcional).
Semana 10	Tema 8 (2 hs de clase). Práctico 5 (1 h de clase opcional).
Semana 11	Tema 9 (2 hs de clase). Consulta Práctico 5 (1 h de clase opcional).
Semana 12	Laboratorio
Semana 13	Laboratorio
Semana 14	Laboratorio
Semana 15	Laboratorio

A3) MODALIDAD DEL CURSO Y PROCEDIMIENTO DE EVALUACIÓN

El curso cuenta con las siguientes instancias de evaluación:

• Realización de ejercicios de práctico durante el curso.
• Prueba escrita o defensa oral, dependiendo de la cantidad de estudiantes que estén realizando el curso
• Trabajo laboratorio final.

Para aprobar la unidad curricular será necesario aprobar cada una de las instancias de evaluación. En caso contrario el curso se pierde. El nivel de aprobación de cada una de las instancias de evaluación es el 50%.

La nota del curso estará compuesta por tres instancias de evaluación:

- Entrega de prácticos (40%)
- Proyecto final (40%)
- Prueba escrita o defensa oral (20%)

A4) CALIDAD DE LIBRE

Esta asignatura no adhiere a la resolución del consejo sobre la Calidad de Libre.

A5) CUPOS DE LA UNIDAD CURRICULAR

No corresponde.

Formato Aprobado por resolución Nº113 del CFI de fecha 04.07.2017

ANEXO B para las carreras de Ingeniería en Computación (plan 97) y Licenciatura em Computación (plan 2012)

B1) ÁREA DE FORMACIÓN

Arquitectura, Sistemas Operativos y Redes de Computadoras.

B2) UNIDADES CURRICULARES PREVIAS

- Examen de Programación 2
- Una de las tres siguientes opciones:
    * Examen de Arquitectura de Computadoras
    * Examen de Sistemas Operativos
    * Curso de Arquitectura de Computadoras y Curso de Sistemas Operativos