Ministerio de Cultura y Educación **(Programa del año 2022)**
Universidad Nacional de San Luis
Facultad de Ciencias Físico Matemáticas y Naturales
Departamento: Informatica
Area: Area II: Sistemas de Computacion

I - Oferta Académica

Materia	Carrera	Plan	Año	Período
() COMPUTACION PARALELA EN GPU	ING. EN COMPUT.	28/12 026/1	2022	1° cuatrimestre
() COMPUTACION PARALELA EN GPU	ING. INFORM.	2- 08/15	2022	1° cuatrimestre
() COMPUTACION PARALELA EN GPU	LIC.CS.COMP.	18/11	2022	1° cuatrimestre
() COMPUTACION PARALELA EN GPU	LIC.CS.COMP.	32/12	2022	1° cuatrimestre

II - Equipo Docente

Docente	Función	Cargo	Dedicación
PICCOLI, MARIA FABIANA	Prof. Responsable	P.Asoc Exc	40 Hs
MIRANDA, NATALIA CAROLINA	Prof. Co-Responsable	P.Adj Exc	40 Hs

III - Características del Curso

Credito Horario Semanal				
Teórico/Práctico	Teóricas	Prácticas de Aula	Práct. de lab/ camp/ Resid/ PIP, etc.	Total
2 Hs	Hs	1 Hs	3 Hs	6 Hs

Tipificación	Periodo
B - Teoria con prácticas de aula y laboratorio	1º Cuatrimestre

Duración			
Desde	Hasta	Cantidad de Semanas	Cantidad de Horas
21/03/2022	24/06/2022	14	75

IV - Fundamentación

El auge de las nuevas tecnologías y la constante demanda de mejores resultados en las aplicaciones hace imprescindible el estudio de nuevas técnicas y metodologías de programación. La enseñanza de la disciplina Sistemas Paralelos tiene como objetivo proveer las herramientas básicas para permitir al estudiante considerar la computación paralela en las nuevas tecnologías o tecnologías dedicadas como son las GPU. Además, le permitiría iniciarse en el campo de la investigación científica y en desarrollos propios de su área de interés.
En el Plan de Estudio de la Licenciatura/Ingeniería se introduce el tema en distintas asignaturas a través de dos tópicos fundamentales: el hardware y el software. La presente asignatura pretende establecer como lograr computación paralela y desarrollar aplicaciones paralelas eficientes y adecuadas a ejecutarse en cualquier arquitectura.

V - Objetivos / Resultados de Aprendizaje

El procesador gráfico, GPU, surge como una alternativa válida para la computación de alto desempeño de aplicaciones de propósito general. Las GPU comienzan a ser una alternativa de bajo costo para el desarrollo de aplicaciones de muy alto rendimiento que tradicionalmente han sido exclusivas de los supercomputadoras. Así mismo, con la aparición del lenguaje CUDA, la programación de las GPU para el desarrollo de aplicaciones de propósito general se ha facilitado enormemente.
Son objetivos de este curso: conocer la potencia actual del procesador gráfico y su utilización para la resolución de

aplicaciones de propósito general. Familiarizarse con las formas más actuales de programación gráfica para propósito general: CUDA y OpenCL. Comparar arquitecturas multi-core y many-core.

VI - Contenidos

Unidad temática 1: GPU: Introducción a GPGPU

• GPU Programación y Arquitectura
• Caracteristicas de la GPU.
• Arquitectura de GPU: N-vidia, Ati, Tesla, Fermi, Kepler
• Pipeline Gráfico
• GPGPU: Computación de Propósito General en GPU.
• Paradigmas de Computación Paralela: Modelo de Memoria Compartida, Paralelismo de Datos.

Unidad temática 2: Programación de GPU a través de CUDA

• Introducción a CUDA
• Modelo de programación de CUDA
• Características Básica.
• Diseño de programas en GPU
• Threading.
• Thread, Bloques, Grid.
• Sincronización de Threads
• Ejemplos.

Unidad temática 3: Modelo de Memoria de GPU

• Modelo de Memoria de GPU.
• Jerarquías de Memoria: Registros, Memoria Compartia, Memoria constante, Texturas, Memoria Global.
• Estructuras de datos básicas.
• Creación de Estructuras de datos en la GPU.
• Ejemplos.

Unidad temática 4: Modelo GPU-CPU

• Modelo CPU-GPU.
• Transferencia de datos GPU-CPU.
• Modelos de programación paralela en GPU: Reducciones, Map, Scan.
• Ejemplos

Unidad temática 5: Performance en GPU

• Optimización de la performance.
• Colisiones de Memoria
• Accesos a Memoria Coallesed.

Unidad temática 6: Otros Lenguajes de Programación de GPU

• Otros modelos de programación GPGPU: OpenCL

VII - Plan de Trabajos Prácticos

Práctico 1: Arquitecturas de GPU- Distintas Generaciones.
Práctico 2: Programación Básica - Memoria Global
Práctico 3: Memoria Shared y de Constante.
Práctico 4: Proyecto Integrador

VIII - Regimen de Aprobación

Acerca del dictado de la Materia:

Las clases tendrán la modalidad teórico-práctica.
Acerca de la regularización de la materia:
El alumno debe:
1- Tener como mínimo un 80% de asistencia a las clases teórico-prácticas.
2- Por cada práctico, la cátedra exigirá la entrega de un ejercicio tipo.
3- Desarrollo de un trabajo práctico evaluativo consistente de:
1. La diseño de una aplicación paralela en GPU.
2. Confección de un informe. Dicho informe deberá: especificar los aspectos teóricos y tener un análisis detallado y justificado de las decisiones tomadas.
3. Exposición del trabajo realizado.

Los puntos 3.1 y 3.2 tendrán tantas recuperaciones como la reglamentaciones vigentes exigen.

Acerca de la aprobación de la materia:

Por promoción: Si la nota obtenida en el trabajo práctico evaluativo obtiene una nota mayor o igual a 7 y el desempeño del alumno durante la cursada fue satisfactorio.
Además se tomará un coloquio oral de la materia.

Por examen regular: Si el alumno regularizó la materia, puede acceder a un examen regular, el cual puede ser oral o escrito y versará sobre los temas del programa.
Dadas las características de la materia y la importancia de los desarrollos prácticos, no se permiten alumnos libres.

IX - Bibliografía Básica

[1] • Buck I. “Gpu computing with nvidia cuda”. ACM SIGGRAPH 2007 courses ACM, 2007. New York, NY, USA.
[2] • Chen W. y Hang H. “H.264/avc motion estimation implementation on compute unified device architecture (cuda)”. In IEEE, editor, IEEE International Conference on Multimedia. 2008.
[3] • Goyal N., Ormont J., Smith R., Sankaralingam K., y Estan C. “Signature matching in network processing using simd-gpu architectures”. In University of Wisconsin. 2008.
[4] • Guil N. y Ujaldón M. “La gpu como arquitectura emergente para supercomputación”. In XIX Jornadas de Paralelismo de Castellon. 2008.
[5] • Joselli M., Zamith M., Clua E., Montenegro A., Conci A., Leal-Toledo R., Valente L., Feijo B., Dórnellas M., y Pozzer C. “Automatic dynamic task distribution between cpu and gpu for real-time systems”. In 11th IEEE International Conference on Computational Science and Engineering. 2008.
[6] • Kerr A.and Diamos G. y Yalamanchili S. “Modeling gpu-cpu workloads and systems”. In 3rd Workshop on GP Computation on Graphics Processing Units. ACM, 2010.
[7] • Kirk, D.,Hwu, W.. “Programming Massively Parallel Processors: A Hands-on Approach”. ISBN: 978-0-12-381472-2. Elsevier. 2010.
[8] • Lieberman M., Sankaranarayanan J., y Samet H. “A fast similarity join algorithm using graphics processing units”. In ICDE 2008. IEEE 24th International Conference on Data Engineering 2008. 2008.
[9] • Lloyd D., Boyd C., y Govindaraju N. “Fast computation of general fourier transforms on gpus”. In IEEE International Conference on Multimedia and Expo. 2008.
[10] • Luebke D. “Cuda: Scalable parallel programming for high-performance scientific computing”. In 5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro, ISBI 2008. 2008.
[11] • Luebke D. H.G. “How gpus work”. EEE Computer, 40(2), 2007.
[12] • Nottingham A. y Irwin B. “Gpu packet classification using opencl: a consideration of viable classification methods”. In Research Conf. of the South African Inst. of Comp. Sc. and Inf. Technologists. ACM, 2009.
[13] • NVIDIA. “Nvidia cuda compute unified device architecture, programming guide version 2.0”. In NVIDIA. 2008a.
[14] • NVIDIA. “Nvidia geforce 8800 gpu architecture overview”. In NVIDIA. 2006.
[15] • NVIDIA. Nvidia geforce gtx 200 gpu architectural overview. In NVIDIA. 2008b.
[16] • Piccoli, María Fabiana. Computación de Alto Desempeño en GPU. Edulp. ISBN 978-950-34-0759-2. 2011.
[17] • Ryoo S., Rodrigues C., Baghsorkhi S., Stone S., Kirk D., y Hwu W. Optimization principles and application performance evaluation of a multithreaded gpu using cuda. In ACM. ACM, 2008.

[18] • Sanders, J., Kandrot, E.. “Cuda by Example: An Introduction to General- Purpose Gpu Programming”. ISBN: 0131387685. Addison-Wesley Professional. 2010.
[19] • Volkov V. “Benchmarking gpus to tune dense linear algebra”. In ACM/IEEE conference on Supercomputing. IEEE Press, Piscataway, NJ, USA, 2008.

X - Bibliografia Complementaria

[1]

XI - Resumen de Objetivos

Conocer la potencia actual del procesador gráfico y su utilización para la solución de aplicaciones de propósito general.
Familiarizarse con las formas más actuales de programación gráfica para propósito general: CUDA y OpenCL.

XII - Resumen del Programa

• GPU: Introducción a GPGPU
• Programación de GPU a través de CUDA
• Modelo de Memoria de GPU
• Modelo GPU-CPU
• Performance en GPU
• Otros Lenguajes de Programación de GPU

XIII - Imprevistos

La materia se dicta de manera presencial, eventualmente se recurirá a la virtualidad mediante plataforma meet, aula virtual classroom y whatsapp.

La cantidad de horas de la materia es 75, estas son las consideradas si bien la cuenta 6x14 da un número mayor, el excedente no es contemplado.

XIV - Otros