	Carrera/ Plan:
TALLER DE PROGRAMACION SOBRE GPU	Licenciatura en Informática Plan 2015/Plan 2012/Plan 2003-07 Licenciatura en Sistemas Plan 2015/Plan 2012/Plan 2003-07 Analista en Tecnologías de la Información y la Comunicación Plan 2017
Año 2021	**Año:** 4to **Régimen de Cursada:** Semestral **Carácter:** Optativa **Correlativas**: Programación Concurrente **Profesor/es:** Adrian Pousa **Hs. semanales**: 9hs

FUNDAMENTACIÓN

Los procesadores gráficos (GPUs) han surgido como una alternativa dentro de los procesadores con múltiples núcleos, por sus características de rendimiento y consumo energético.

El uso de GPUs, tanto en computación de alto desempeño como en aplicaciones de propósito general comienza a ser una alternativa de bajo costo para el desarrollo de aplicaciones de muy alto rendimiento que tradicionalmente han sido exclusivas de los clusters de multicores y supercomputadoras. En este contexto, la metodología e implementación de aplicaciones es un tema de gran interés actual.

Son objetivos de este curso: profundizar el conocimiento de las arquitecturas tipo GPU y su programación, comparar su rendimiento con arquitecturas convencionales, analizar los modelos de resolución de problemas específicos e introducir conceptos de consumo y green computing a partir de la utilización de GPUs.

OBJETIVOS GENERALES

Son objetivos de este curso:

- Profundizar el conocimiento de las arquitecturas tipo GPU y su programación.
- Comparar su rendimiento con arquitecturas convencionales.
- Analizar los modelos de resolución de problemas específicos.
- Introducir conceptos de consumo y green computing a partir de la utilización de GPUs.

COMPETENCIAS

- LI-CE1- Planificar, dirigir, realizar y/o evaluar proyectos de especificación, diseño, implementación, verificación, validación, puesta a punto, mantenimiento y actualización para arquitecturas de sistemas de procesamiento de datos, con capacidad de incorporar aspectos emergentes del cambio tecnológico.
- LI-CE6- Controlar las normas de calidad en el software o software integrado a otros componentes. Capacidad de evaluación de perfomance de sistemas de software y sistemas que integren hardware y software.
- LS-CE5- Establecer métricas y normas de calidad y seguridad de software, contralando las mismas a fin de tener un producto industrial que respete las normas nacionales e internacionales. Control de la

especificación formal del producto, del proceso de diseño, desarrollo, implementación y mantenimiento. Establecimiento de métricas de validación y certificación de calidad. Capacidad de evaluación de perfomance de sistemas de software y sistemas que integren hardware y software.
- LS-CE9- Analizar y evaluar proyectos de especificación, diseño, implementación, puesta a punto, mantenimiento y actualización de sistemas de procesamiento de datos, con capacidad de incorporación de tecnologías emergentes del cambio tecnológico.

CONTENIDOS MINIMOS (de acuerdo al Plan de Estudios)

- GPU: Introducción a GPGPU
- Arquitecturas GPU - Modelo GPU-CPU.
- Modelo de Programación GPU - Resolución de aplicaciones - Métricas.
- Modelo y jerarquía de Memoria de GPU.
- Optimizaciones sobre GPU.
- MultiGPUs. Integración con otras arquitecturas. Arquitecturas Híbridas.

PROGRAMA ANALÍTICO

Unidad 1: GPU: Introducción a HPC y GPGPU

Introducción al cómputo de altas prestaciones (HPC). Concepto de sistema paralelo. Hardware y Software paralelo. Clasificación de Hardware paralelo: Taxonomía de Flynn y Clasificación según el modelo de memoria. Jerarquías de memoria. Software paralelo: paralelismo funcional o de control y paralelismo de datos, características de las aplicaciones (CPU, Memoria, Entrada/Salida, Fases), herramientas de desarrollo. Introducción a las arquitecturas GPU y su uso en HPC. Concepto GPGPU: Computación de Propósito General en GPU.

Unidad 2: Arquitecturas GPU - Modelo GPU-CPU

Evolución de las GPUs. Arquitecturas Nvidia. Otras arquitecturas manycore. Modelo de interacción GPU-CPU. Introducción a la planificación de hilos en GPU Nvidia - Concepto de Grid, Bloque, Thread y Warp. Rendimiento y consumo de las arquitecturas GPU según Top500 y Green500.

Unidad 3: Modelo de Programación GPU - Resolución de aplicaciones

Modelo de programación en GPU. Relación con SIMD, modelo SIMT. Modelo de programación CUDA. Concepto de Host y Device. Identificadores. Tipos de datos. Definición de Constantes. Variables: alcance y tiempo de vida. Gestión de memoria, copia explicita CPU-GPU (transferencias H2D) y GPU-CPU (transferencias D2H). Gestión de Hilos: Grid, Bloques, Threads. Dimensiones: 1D, 2D y 3D. Kernel, llamados Síncronos y Asíncronos. Funciones. Identificadores de Threads y Bloques.Planificación de Threads. Sincronización de Threads. Diseño de programas en GPU. Estudio experimental de casos. Métrica de rendimiento: speedup. Análisis de rendimiento. Aceleración en GPU con respecto a CPU.

Unidad 4: Modelo y jerarquía de Memoria de GPU

Modelo de Memoria de GPU. Jerarquía de Memoria: Registros, Memoria Compartida, Memoria de constantes, Memoria de Texturas, Memoria Global. Memorias Cache: Constantes, Texturas, Nivel 1, Nivel 2. Patrones de Acceso a Memoria Global, relación entre segmentos y cantidad de transacciones. Patrones de Acceso a Memoria Compartida, bancos de memoria, conflicto de bancos, accesos sin conflictos. Concepto de Acceso Coalescente. El problema de la latencia.

Unidad 5: Optimizaciones

Coalescencia y prefetching. Divergencia. Mezcla y granularidad de instrucciones. Ocupación y asignación de recursos.

Unidad 6: CUDA Streams

Impacto de las transferencias H2D y D2H. Ocultamiento de la latencia. CUDA Streams. Concurrencia a nivel Kernel y a nivel de memoria. Copia de memoria asíncrona. Gestión de eventos.

Unidad 7: Modelo Híbrido

Máquinas multi-GPUs. Arquitectura Híbridas: Modelo Multicore-GPU, Modelo Cluster-GPU y cluster de GPUs, Modelo Multicore-Cluster-GPUs. Heterogeneidad. Distribución de carga.

BIBLIOGRAFÏA

Grama A, Gupta A, Karypis G, Kumar V. “Introduction to parallel computing”. Second Edition. Pearson Addison Wesley, 2003.

Jason Sanders, Edward Kandrot “CUDA by Example An Introduction to General Purpose GPU Programming”. NVIDIA Corporation, Addison Wesley, 2011.

M. F. Piccoli, “Computación de Alto Desempeño utilizando GPU”. XV Escuela Internacional de Informática. Editorial Edulp, 2011.

John Cheng, Max Grossman, Ty McKercher, “Professional CUDA C Programming”, John Wiley & Sons, 2014.

David B. Kirk, Wen-mei W. Hwu, “Programming Massively Parallel Processors - A Hands-on Approach” 3ed, NVIDIA Corporation and Wen-mei W. Hwu, Elsevier, 2017.

METODOLOGÍA DE ENSEÑANZA

Modalidad presencial

La asignatura sigue una modalidad taller en la cual se alternan clases teóricas con prácticas experimentales.

Durante el desarrollo de la asignatura no se tomará asistencia. La existencia de un control de asistencia es con fines estadísticos.

Las clases teóricas introducen los conceptos teóricos de la asignatura aplicables en las prácticas experimentales.

Las prácticas experimentales se desarrollan la Sala de Cómputo de Postgrado (por la disponibilidad de arquitecturas paralelas tipo GPU y la configuración de estas en modo cluster) y equipamiento especial del III-LIDI accedido de forma remota.

Modalidad semi-presencial

Dada la no obligatoriedad de las clases, los alumnos en modalidad semi-presencial pueden seguir los temas por el entorno IDEAS y asistir a las consultas que se fijen para los alumnos presenciales.

Se hace notar que por la característica de las tareas experimentales, el alumno deberá tener acceso al menos a un modelo de arquitectura paralela que incluya GPUs para poder realizar los trabajos que se requieren durante el curso. Es recomendable que asistan a las clases prácticas experimentales para el desarrollo de los conocimientos que se adquieren en dichas clases.

Redictado

Dada las características de la materia y la experiencia de años anteriores la cátedra no ve necesidad alguna de realizar un redictado.

Modalidad virtual de situación excepcional

En caso de continuar la situación excepcional del año 2020 se seguirá una modalidad virtual. Las clases y las explicaciones relacionadas a las prácticas experimentales se dictarán mediante alguna plataforma virtual (Webex o similar) y el material será publicado en la plataforma IDEAS. Las consultas podrán hacerse sincrónicamente a través de la plataforma virtual o asincrónicamente a través de la mensajería de IDEAS. Las prácticas experimentales se llevarán a cabo sobre equipamiento especial, provisto por la cátedra, accedido de forma remota.

EVALUACIÓN

Modalidad presencial

Para obtener la aprobación de cursada de la asignatura los alumnos deben aprobar todas las entregas de los diferentes trabajos experimentales, estas entregas pueden ser en grupos de 2 personas. Cada trabajo es acompañado por un coloquio y puede tener solo una re-entrega.

Además de las entregas, los alumnos deben aprobar un examen parcial. Las instancias de examen consisten en una fecha y dos recuperatorios.

Para la aprobación de la nota final se computa el promedio de la nota de los trabajos y sus respectivos coloquios con la nota del parcial.

La nota del final será válida por un semestre posterior a la finalización de la materia pasado ese período el alumno deberá revalidar la nota del final mediante un examen escrito.

Modalidad semi-presencial y Evaluación por promoción

Deben cumplir con los mismos requisitos que los alumnos en modalidad presencial.

Modalidad virtual de situación excepcional

Se cumplirá con los mismos requisitos que la modalidad presencial. Los coloquios y el exámen parcial se realizarán empleando alguna plataforma virtual.

CRONOGRAMA DE CLASES Y EVALUACIONES

Clase	Fecha	Contenidos/Actividades
1	**Semana del 16/8**	Unidad 1
2	**Semana del 23/8**	Unidad 2
3	**Semana del 30/8**	Unidad 3
4	**Semana del 6/9**	Actividades y consultas Prácticas. Presentación de Trabajo práctico a entregar.
5	**Semana del 13/9**	Entrega y corrección de Trabajos Prácticos.
6	**Semana del 20/9**	Unidad 4
7	**Semana del 27/9**	Unidad 5
8	**Semana del 4/10**	Actividades y consultas Prácticas. Re- Entrega y corrección.
9	**Semana del 11/10**	Actividades y consultas Prácticas. Presentación de Trabajo práctico a entregar.
10	**Semana del 18/10**	Entrega y corrección de Trabajos Prácticos.
11	**Semana del 25/10**	Unidad 6
12	**Semana del 1/11**	Actividades y consultas Prácticas. Re- Entrega y corrección.
13	**Semana del 8/11**	Actividades y consultas Prácticas. Presentación de Trabajo práctico a entregar.
14	**Semana del 15/11**	Entrega y corrección de Trabajos Prácticos.
15	**Semana del 22/11**	Parcial 1ra Fecha – Corrección y muestra
16	**Semana del 29/11**	Consultas Prácticas. 1er Recuperatorio del parcial – Corrección y muestra
17	**Semana del 6/12**	Consultas Prácticas. 2do Recuperatorio del parcial – Corrección y muestra
18	**Semana del 13/12**	Re- Entrega y corrección de Trabajos Prácticos.

Evaluaciones previstas	Fecha
Parcial 1ra fecha	Semana 22/11 – Fecha tentativa 24/11/2020
Parcial 2da fecha	Semana 29/11 – Fecha tentativa 01/12/2020
Parcial 3ra fecha	Semana 6/12 – Fecha tentativa 10/12/2020

Contacto de la cátedra (mail, sitio WEB, plataforma virtual de gestión de cursos):

Plataforma virtual: https://ideas.info.unlp.edu.ar
Web: http://weblidi.info.unlp.edu.ar/catedras/tallerGPU/
Mail: apousa@lidi.info.unlp.edu.ar

Firma del/los profesor/es