DENOMINACIÓN
COMPUTACIÓN PARALELA

FUNDAMENTACIÓN
Esta asignatura corresponde al campo de las herramientas básicas en el área espacial, estando presente en todos los trayectos sugeridos de la Maestría Aeroespacial. Presenta como lineamientos generales conocer y aprender a desarrollar las tres dimensiones de paralelismo que actualmente posee una arquitectura de microprocesador. La demanda de personal calificado se incrementa permanentemente debido al crecimiento de la actividad espacial año a año situación que obliga a poner énfasis en la formación de los profesionales necesarios para atenderla.

La mayor complejidad de los sistemas espaciales requiere aprovechar de la tecnología de los microprocesadores modernos que contienen un alto nivel de paralelismo. Para poder utilizar eficientemente estas arquitecturas se deben conocer los modelos de ejecución y las formas de sacar provecho a este paralelismo.

OBJETIVOS DE LA ASIGNATURA
- Presentar las tres dimensiones de paralelismo de una arquitectura de microprocesador: paralelismo de instrucciones (ILP), de datos (DLP) y de hilos (TLP), tanto en sus variantes de CPU como de GPU.
- Comprender las soluciones de compromiso de cada una de estas arquitecturas para obtener alto desempeño tanto en cálculo como en acceso a memoria.
- Saber discernir si un proceso está realizando un uso adecuado de todas las capacidades de la máquina.
- Adquirir un nivel formativo que facilite la incorporación del profesional a grupos de trabajo dedicados a la investigación y a la aplicación industrial en áreas de la especialidad.

CONTENIDOS
Unidad 1 Introducción.
Escalado. Leyes de: Amdahl, Gustafson, Little. Eficiencia. Factores que degradan el desempeño: inanición, latencia, sobrecarga, contención. Paralelización: descomposición en tareas, orden y agrupamiento de tareas, descomposición de datos, datos compartidos. Sincronización: condiciones de carrera, instrucciones atómicas. Primitivas de sincronización: mutexes, spinlocks, semáforos, barreras y fences. Predicción de desempeño: modelo roofline. Medición de desempeño.
Unidad 2 CPU
Paralelismo de instrucción (ILP): pipelining, procesadores superescalares, ejecución fuera de orden, SMT. Memoria: jerarquía y asociatividad de cache, alineamiento de memoria, algoritmos cache-aware y cache-oblivious. Memoria virtual: efectos de la TLB en el desempeño. Memoria distribuída: NUMA, coherencia de cache. Afinidad de memoria y pinning de hilos a cores. Vectorización: unidades SIMD, SSE intrinsics, técnicas de vectorización. OpenMP: constructores work-sharing, atributos para compartir datos, planificadores, sincronización, entorno de ejecución, compilación. Aplicaciones: extensiones ISA específicas para aplicaciones, bibliotecas para HPC.
Unidad 3 GPU
Arquitectura interna. Limitaciones de la GPGPU: serialización de saltos, ocultamiento de la latencia, ocupación. Jerarquía de memoria, cache de software vs. cache de hardware, unidades de textura. CUDA: mapeo hilo-dato, lanzamiento de kernels, comunicación host-device, sincronización, contadores de desempeño y profiling, manejo de errores, compute capabilities, PTX ISA. Optimización: aumento de la granularidad de los hilos, uso efectivo de la memoria

compartida, código sin saltos, double buffering, reducción del uso de registros, aritmética de precisión mixta, cómo evitar instrucciones atómicas. Ejemplos de Algoritmos GPU: reducción, scan segmentado, compactación de streams y sus usos. Bibliotecas: CUBLAS, CUFFT, CUSPARSE, Thrust, CUDPP, CUB.

ACTIVIDADES PRÁCTICAS

El alumno deberá elegir un programa de computación numérica intensiva, que será paralelizado de 4 formas:

1. ILP, cache-aware.
2. SIMD (instrucciones vectoriales).
3. Multicore (típicamente OpenMP para CPU).
4. Manycore (típicamente CUDA para GPU).

Se desarrollará en los laboratorios de computación y también de forma no-presencial, estimando igual tiempo de práctico (60hs) que de trabajo en la casa (60hs). La supervisión será en las 60hs de práctico.

Se deberá entregar un informe final donde se comparen las mejoras obtenidas. En el primer punto se deberá analizar la mejor utilización de las unidades de ejecución, y la mejora en las tasas de cache-hit. En el segundo caso, la mejora que se obtuvo al operar de manera vectorial sobre los datos y la estrategia de paralelización utilizada, así como un análisis de scaling respecto al ancho de la unidad vectorial (AVX vs AVX512). Para multicore CPU además de analizar el scaling con respecto a la cantidad de cores, se deberá informar sobre los efectos de la afinidad de memoria-cpu. Finalmente, para manycore GPU se harán análisis de weak-scaling y de utilización del ancho de banda de memoria y potencia de cálculo respecto al pico teórico.

METODOLOGÍA

La metodología de enseñanza para esta asignatura se plantea en el marco del dictado de clases teórico/prácticas.

El sistema de enseñanza es de carácter teórico-práctico, con preeminencia del método deductivo (de lo general a lo particular) al tratar la faz teórica de los temas listados en los contenidos. En la medida de lo posible, siempre se intentará lograr que las clases por su contenido y modalidad de dictado estimulen la participación de los maestrandos.

La parte teórica de las clases tiene carácter expositivo, donde el docente presenta las definiciones, conceptos y formulaciones computacionales. La parte práctica presenta una mayor interacción, debido a que se aplica un formato de exposición dialogada, guiando a los alumnos a realizar análisis deductivos para poder hallar las soluciones a los problemas planteados, usando los conceptos desarrollados en la parte teórica. Se destaca que las clases están formalmente divididas en teóricas y prácticas.

La estructura de dictado de la asignatura consiste en dos clases semanales. Además, los docentes establecen un horario de consulta por fuera del horario de clases formal, el cual tiene una extensión adecuada en función de la cantidad de maestrandos inscriptos en la asignatura.

Se espera que la metodología aplicada desarrolle en el maestrando las competencias para:

- Identificar las soluciones de compromiso de cada una de las tres arquitecturas presentadas para obtener alto desempeño tanto en cálculo como en acceso a memoria.
- Adecuar un proceso computacional para realizar un uso adecuado de todas las capacidades de la máquina.
    - Aplicar correctamente los conceptos y métodos adecuados para la resolución de problemas.
- Adquirir un nivel formativo que facilite la incorporación a grupos de trabajo dedicados a la investigación y a la aplicación industrial en áreas de la especialidad.
    - Desarrollar análisis crítico y criterio analítico sobre planteo y solución de problemas relacionados con el uso de computación paralela.

Además, su busca que el maestrando adquiera competencias de carácter por un lado actitudinal, como el cumplimiento de responsabilidades y obligaciones y tener participación activa en las actividades prácticas, y por otro aptitudinal, como la identificación de problemas y la organización del tiempo y tareas.

EVALUACIÓN

Se deberá presentar los 4 prácticos y realizar un proyecto que tome un problema del dominio de trabajo en el posgrado del maestrando y aplicar las técnicas aprendidas, presentando un informe y los códigos correspondientes.

Las condiciones para la promoción de la asignatura son:

- Presentar y aprobar con nota no inferior a siete (7) en una escala de cero (0) a diez (10) cada uno de los prácticos y proyectos que se exijan durante el desarrollo de los trabajos prácticos.

Los maestrandos que cumplan con el 50% de las exigencias referidas a los proyectos serán considerados regulares. Los demás estarán libres.

La nota final corresponderá al promedio ponderado de los trabajos prácticos y proyectos.

CARGA HORARIA

Modalidad	Carga Teórica	Carga Práctica	TOTAL
Presencial	**20**	**40**	**60**
A distancia	**-**	**-**	**-**
TOTAL	**20**	**40**	**60**

BIBLIOGRAFÍA

Chapman, Jost, van der Pas, Using OpenMP: Portable Shared Memory Parallel Programming, 2007.

Hennessy, Patterson, Computer Architecture a Quantitative Approach, 5th edition, Morgan Kaufmann, 2011.

Hennessy, Patterson, Computer Organization and Design: the Hardware / Software Interface, 5th edition, Morgan Kaufmann, 2013.

Kirk, Hwu, Programming Massively Parallel Processors, 2nd edition, 2012.

NVIDIA Inc., CUDA C Programming Guide, versión CUDA 9.1, 2017.

NVIDIA Inc., CUDA C Best Practices, versión CUDA 9.1, 2017.

NVIDIA Inc., PTX ISA 6.1, versión CUDA 9.1, 2017.