



RESOLUCION HCD N°326/11

VISTO:

El pedido realizado por los Lic. Nicolás Wolovick y Carlos Bederian para que se incorpore la materia "Computación Paralela" como Optativa de la Licenciatura en Ciencias de la Computación; y

CONSIDERANDO:

Que se cuenta con el acuerdo de la Comisión Asesora de Computación;

Que es conveniente agregar a la nómina de materias optativas, aprobada por Res. HCD N° 207/02, la asignatura que se propone;

Que mediante Resolución HCS N° 122/02 se ha delegado en este Cuerpo la facultad de modificar la nómina de materias optativas del Plan de Estudios de la Licenciatura en Ciencias de la Computación;

EL HONORABLE CONSEJO DIRECTIVO DE LA
FACULTAD DE MATEMÁTICA, ASTRONOMÍA Y FÍSICA
R E S U E L V E

ARTÍCULO 1°: Hacer lugar a lo solicitado por los Lics. Wolovick y Bederian y, en consecuencia, modificar la nómina de materias optativas del Plan de Estudios de la Licenciatura en Ciencias de la Computación, incorporando a la misma la materia "Computación Paralela".

ARTÍCULO 2°: Fijar como programa, correlativas y carga horaria de la materia, los detallados en el Anexo que forma parte de la presente Resolución.

ARTÍCULO 3°: En cumplimiento con lo establecido en el Artículo 2° de la Res. HCS N°122/02, remítase a la Secretaría de Asuntos Académicos de la Universidad la presente resolución para su conocimiento y efectos.

ARTÍCULO 4°: Comuníquese y archívese.

DADA EN LA SALA DE SESIONES DEL HONORABLE CONSEJO DIRECTIVO DE LA FACULTAD DE MATEMÁTICA, ASTRONOMÍA Y FÍSICA, A DOCE DÍAS DEL MES DE DICIEMBRE DE DOS MIL ONCE.

pk


Dra. ESTHER GALINA
VICEDECANA
a/c Secretaría General
Fa.M.A.F


Dr. FRANCISCO A. TAMARIT
DECANO
Fa.M.A.F

11/11



ANEXO A RESOLUCIÓN HCD N°326/11

MATERIA OPTATIVA	CORRELATIVAS			CARGA HORARIA
	PARA CURSAR		PARA RENDIR	
	REGULARIZADA	APROBADA	APROBADA	
Computación Paralela	Sistemas Operativos	Algoritmos y Estructura de Datos I	Sistemas Operativos	120 hs.
	Arquitectura de Computadoras	Organización del Computador	Arquitectura de Computadoras	

Régimen de Cursado: Semestral.

FUNDAMENTACIÓN Y OBJETIVOS

Este curso optativo para la currícula de la Lic. en Ciencias de la Computación apunta a formar estudiantes en técnicas modernas para el desarrollo de código eficiente para Computación de Alto Desempeño (CAD) en arquitecturas de muchos núcleos, cubriendo las tecnologías CPU y GPU. En ambos paradigmas se presenta la arquitectura, un paradigma de programación moderno y las técnicas particulares para obtener el máximo beneficio de cada una, sin sacrificar corrección. También incluiremos un paradigma híbrido que cubre ambas arquitecturas. El curso también intenta desarrollar en los estudiantes una mirada crítica de la batalla que se está librando entre las arquitecturas de CPU y GPU.

CONTENIDOS

Introducción

- Escalado, ley de Amdahl, 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.



- Sincronización: condiciones de carrera, instrucciones atómicas. Primitivas de sincronización: mutexes, spinlocks, semáforos, barreras y fences.
- Medición de desempeño.

CPU

1. Paralelismo de instrucción: pipelining, procesadores superescalares, ejecución fuera de orden, SMT.
2. 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.
3. Vectorización: unidades SIMD, SSE intrinsics, técnicas de vectorización.
4. OpenMP: constructores work-sharing, atributos para compartir datos, planificadores, sincronización, entorno de ejecución.
5. Aplicaciones: extensiones ISA específicas para aplicaciones, bibliotecas para HPC.

GPU

1. Arquitectura.
2. Limitaciones de la GPGPU: serialización de saltos, ocultamiento de la latencia, ocupación.
3. Jerarquía de memoria, cache de software versus cache de hardware, unidades de textura.
4. CUDA: mapeo hilo-dato, lanzamiento de kernels, comunicación host-device, sincronización, contadores de desempeño y profiling, manejo de errores, compute capabilities, PTX ISA.
5. 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.
6. Algoritmos GPGPU: reducción, scan segmentado, compactación de streams y sus usos.
7. Bibliotecas: CUBLAS, CUFFT, CUSPARSE, Thrust, CUDPP.

Computación heterogénea

1. OpenCL: introducción y mapeo desde CUDA, ejecución en CPUs y GPUs, acercamientos híbridos.



METODOLOGÍA DE TRABAJO Y EVALUACIÓN

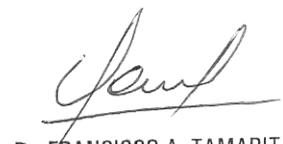
El curso será una materia optativa regular de 120hs, dividida en 14 semanas. Se dedicarán 2hs por semana de exposición frente al curso, 2hs por semana de laboratorio y 4hs por semana de trabajo dirigido.

BIBLIOGRAFÍA

1. T. G. Mattson , B. A. Sanders , B. L. Massingill, Patterns for Parallel Programming, 2004.
2. B. Chapman, G. Jost, R. van der Pas, Using OpenMP: Portable Shared Memory Parallel Programming, 2007.
3. J. Sanders, E. Kandrot, CUDA by Example: An Introduction to General-Purpose GPU Programming, 2010.
4. D. B. Kirk, Wen-mei W. Hwu, Programming Massively Parallel Processors, 2010.
5. NVIDIA Inc., CUDA Programming Guide, CUDA Best Practices and PTX ISA, version 4.0, 2011.
6. J. Hennessy, D. Patterson, Computer Architecture a Quantitative Approach, 4th edition, Morgan Kaufmann, 2006.
7. J. Hennessy, D. Patterson, Computer Organization and Design : the Hardware / Software Interface, 4th edition, Morgan Kaufmann, 2008.



Dra. ESTHER GALINA
VICEDECANA
a/c Secretaría General
Fa.M.A.F.



Dr. FRANCISCO A. TAMARIT
DECANO
Fa.M.A.F.