DENOMINACIÓN
COMPUTACIÓN PARALELA

CARGA HORARIA

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

OBJETIVOS
Que el estudiante comprenda las tres dimensiones de paralelismo que actualmente posee 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.
Al final de la materia los estudiantes deben ser capaces de adaptar programas a fin de utilizar estas tres dimensiones del paralelismo, tanto en CPU como en GPU.

CONTENIDOS

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.

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.

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 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.
La evaluación se dá con un coloquio colectivo en el práctico la clase siguiente a la finalización del práctico.

MODALIDAD DE EVALUACIÓN

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

BIBLIOGRAFÍA

- B. Chapman, G. Jost, R. van van der Pas, Using OpenMP: Portable Shared Memory Parallel Programming, 2007.
- D. B. Kirk, Wen-mei W. 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.
- J. Hennessy, D. Patterson, Computer Architecture a Quantitative Approach, 5th edition, Morgan Kaufmann, 2011.
- J. Hennessy, D. Patterson, Computer Organization and Design: the Hardware / Software Interface, 5th edition, Morgan Kaufmann, 2013.