PROGRAMA DE CURSO DE POSGRADO

TÍTULO: Computación Paralela	
AÑO: 2014	**CUATRIMESTRE: primero**
CARGA HORARIA: 120	**No. DE CRÉDITOS: 3**
CARRERA/S: Computación, Matemática, Astronomía, Física.	
Docente encargado: Dr. Nicolás Wolovick **Docente de práctico: Lic. Carlos Bederián**	

PROGRAMA

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.
- 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 versus 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.

Computación heterogénea

- Utilización de sistemas runtime para la paralelización heterogénea (CPU+GPU) de algoritmos. Ejemplos de uso. Bibliotecas de álgebra lineal heterogéneas.

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, version CUDA 5.5, 2013.
- NVIDIA Inc., CUDA C Best Practices, version CUDA 5.5, 2013.
- NVIDIA Inc., PTX ISA 3.2, version CUDA 5.5, 2013.
- 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.
- INRIA, StarPU Handbook, version 1.1.0.

MODALIDAD DE LA EVALUACIÓN

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 entregará 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, asi como un mínimo análisis de *scaling* respecto al ancho de la unidad vectorial (SSE4 vs. AVX). 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.