



UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA

**SISTEMAS DISTRIBUIDOS Y
PARALELOS**

Año 2021

Carrera/ Plan:

Ingeniería en Computación Plan 2008/2011

Año: 5to

Regimen de Cursada: *Semestral*

Carácter: Obligatoria

Correlativas: Concurrencia y Paralelismo

Profesor: Adrian Pousa

Docentes Auxiliares: -

Hs. semanales: 6 hs.

Hs. totales: 96 hs.

Fundamentación

La evolución tecnológica de los procesadores ha impuesto el procesamiento paralelo. Formar al alumno (que ya tiene conocimientos previos de Concurrencia y sus aplicaciones) en los fundamentos de los sistemas paralelos, los paradigmas de programación paralela y las métricas de rendimiento asociadas resulta un aporte fundamental para el futuro profesional. Esta tarea de formación se combina con trabajo experimental sobre sistemas paralelos concretos, disponibles en la Facultad.

Objetivos generales:

Caracterizar los problemas de procesamiento paralelo desde dos puntos de vista: la arquitectura física y los lenguajes de programación, poniendo énfasis en la transformación de algoritmos secuenciales en paralelos. Describir los modelos de cómputo paralelo y los paradigmas de programación paralela. Estudiar las métricas de performance asociadas al paralelismo, así como modelos de predicción de performance orientados a diferentes arquitecturas multiprocesador. Plantear casos concretos de procesamiento paralelo, resolubles sobre distintas arquitecturas multiprocesador.



Contenidos Mínimos:

- Arquitecturas de procesamiento paralelo.
- Diseño de algoritmos paralelos.
- Programación de algoritmos paralelos: Modelo de memoria compartida. Modelo de memoria distribuida. Modelos híbridos.
- Paradigmas de resolución de sistemas paralelos.
- Métricas.

Programa Analítico

Unidad 1: Conceptos básicos

Concurrencia, Paralelismo y Cómputo distribuido. Objetivos del procesamiento paralelo. Concepto de Sistema Paralelo. Software de un Sistema Paralelo. Hardware de un Sistema Paralelo. Motivación de los Sistemas Paralelos. Áreas de aplicación. Limitaciones actuales de hardware. Ley de Moore. Conceptos de paralelismo implícito y explícito. Paralelismo implícito Conceptos de Segmentación/Pipelining y división funcional. Transparencia. Procesador Superescalar. Paralelismo explícito. Etapas en el diseño de algoritmos paralelos: Descomposición, Asignación/Aglomeración, Orquestación, Mapeo. Herramientas de desarrollo. Herramientas basadas en el modelo de memoria distribuida. Herramientas basadas en el modelo de memoria compartida. Impacto del procesamiento paralelo sobre lenguajes de programación y los sistemas operativos. Problemas de la programación paralela. Concepto de High Performance Computing (HPC).

Unidad 2: Arquitecturas Paralelas

Clasificación de arquitecturas paralelas. Por Organización lógica: por mecanismo de control (Taxonomía de Flynn: SISD. SIMD. MISD. MIMD) y por modelos de comunicación, arquitecturas de memoria compartida y distribuida. Por Organización física: por organización del espacio de direcciones, por red de interconexión y por la granularidad de los procesadores.

Evolución y Tendencias en las arquitecturas paralelas. Multiprocesadores, Clusters, Multiclusters, Grid, Multicores (homogéneos, heterogéneos y asimétricos), coprocesadores GPU-Manycores, Cloud para HPC. Top 500 y Green 500.

Redes de comunicación en sistemas paralelos y distribuidos orientadas a HPC: Ethernet, Infiniband, Myrinet.



Unidad 3: Sistema de memoria

Sistema de memoria. Limitaciones. La importancia de la memoria caché. Conceptos asociados a la memoria cache (Estados, Niveles, Coherencia) Principio de localidad: Localidad Temporal y Espacial.

Principio de localidad aplicado a la multiplicación de matrices

Minimizando el espacio de memoria. Caso de estudio matrices triangulares.

Conceptos de contadores hardware para evaluar el comportamiento del sistema de memoria.

Unidad 4: Principios de diseño de algoritmos paralelos

Paralelismo explícito. Etapas de diseño de algoritmos paralelos: Descomposición, Asignación/Aglomeración, Orquestación, Mapeo.

Etapas de descomposición. Descomposición en tareas. Tareas estáticas y dinámicas Volumen de tareas. Descomposición funcional. Descomposición de datos. Dependencias. Granularidad. Técnicas de descomposición de datos: recursiva, basada en datos (entrada, salida, entrada/salida, datos intermedios), exploratoria y especulativa. Técnica de descomposición híbrida.

Etapas de asignación/aglomeración. Reducción del volumen de tareas. Asignación, balance de carga. Aglomeración, localidad.

Etapas de orquestación. Elección de herramientas. Modelo de sincronización. Modelos y patrones de comunicación.

Etapas de Mapeo. Mapeo estático. Estrategias de mapeo estático: basado en datos (arrays y grafos), en particiones de tareas y jerárquico. Mapeo dinámico. Estrategias de mapeo dinámico: centralizados y distribuidos. Mapeo dinámico y balance de carga.

Diseño sobre arquitecturas modernas. Heterogeneidad e integración: arquitecturas, sistemas operativos, herramientas y compiladores. Afinidad.

Unidad 5: Programación de algoritmos paralelos. Modelo de memoria compartida

Concepto de thread. Herramientas de desarrollo Posix Threads (Pthreads) y OpenMP. Pthreads: Gestión de hilos, Sincronización, Afinidad. OpenMP: modelo Fork-Join, Estructura de control paralela, Distribución de trabajo entre hilos, Gestión de ambiente de datos, Sincronización, Funciones y variables de ambiente, Afinidad.

Resolución de aplicaciones específicas.

Unidad 6: Programación de algoritmos paralelos. Modelo de memoria distribuida

Principios de la comunicación/sincronización por pasaje de mensajes. Herramientas de desarrollo PVM y MPI. MPI: Funcionamiento y Estructura de programa, Comunicación Punto a Punto y Colectiva. Ocultamiento de la latencia.

Resolución de aplicaciones específicas.



Unidad 7: Programación de algoritmos paralelos. Modelo híbrido

Arquitecturas paralelas híbridas y desafíos en la programación paralela. Modelos híbridos: MPI-Pthreads y MPI-OpenMP. Compilación y Modularidad.

Resolución de aplicaciones específicas.

Unidad 8: Métricas

Importancia de las métricas en sistemas paralelos. Métricas de rendimiento, speedup, speedup absoluto y relativo. Speedup lineal, Speedup perfecto, Speedup superlineal. Eficiencia. Overhead. Balance de carga. Escalabilidad en los sistemas paralelos: Ley de Amdahal, Ley de Gustafson, Definición de Escalabilidad, Modelo de Isoeficiencia.

Métricas en arquitecturas heterogéneas. Concepto de potencia de cómputo relativa (PCR). Limitación del speedup.

Unidad 9: Paradigmas

Paradigmas de programación paralela: Paralelismo de Datos SIMD/SPMD, Divide y Vencerás, Pipelines, Algoritmos sistólicos, Master-Worker, Task pools. Modelos híbridos.

Ejemplos de resolución de aplicaciones específicas.

Unidad 10: Introducción a la programación sobre GPUs

Era Manycore - Introducción al concepto de GPGPU. Características paralelas de las GPUs.

Arquitectura Nvidia: Sistema de procesamiento y Sistema de memoria. Modelo de programación Nvidia CUDA: Declaración de variables, Gestión de la memoria global, Gestión de hilos, Kernel, Modularidad, Variables Built-in y Thread ID, Planificación. Optimizaciones Nvidia CUDA.

Ejemplos de resolución de aplicaciones específicas.



Metodología de Enseñanza

La asignatura se estructura con clases teóricas y clases prácticas experimentales. Las clases se dictarán de forma virtual mediante la plataforma Webex u otro medio similar. No son obligatorias pero se recomienda asistir para un mejor aprendizaje de los contenidos, considerando la fuerte vinculación entre las clases teóricas, prácticas y evaluaciones. La existencia de un control de asistencia es a fines estadísticos.

- Las clases teóricas introducen los conceptos teóricos de la asignatura aplicables en las clases prácticas.
- Las clases prácticas experimentales se desarrollan sobre equipamiento especial provisto por la cátedra que es accedido de forma remota. Estas clases consisten de un conjunto de problemas que deben desarrollar los alumnos en las arquitecturas disponibles. Durante los días de práctica pueden incluirse explicaciones de práctica introductorias al trabajo en la sala y relacionadas a la utilización del equipamiento.
- Las consultas se realizarán de forma sincrónica mediante la plataforma virtual. Eventualmente, se responderán consultas de forma asincrónica mediante el entorno IDEAS cuando involucren dudas puntuales de menor complejidad.

Evaluación:

Para obtener la aprobación de la cursada los alumnos deben aprobar todas las entregas de los diferentes trabajos experimentales, un coloquio asociado a estas entregas y una evaluación.

Las entregas pueden realizarse en grupos de a lo sumo dos personas. Cada entrega puede tener sólo una re entrega.

El coloquio consiste en una defensa oral de las entregas que se realizará de forma sincrónica mediante la plataforma virtual.

La evaluación consiste en un examen que permita evaluar los conocimientos adquiridos en las clases teóricas y en las prácticas experimentales. Se aprobará con nota 6 o más (en una escala de 1 a 10) y tendrá dos instancias recuperatorias para quienes no alcancen esta nota.

Para acceder a la promoción los alumnos deben cumplir los siguientes requisitos:

- Estar inscriptos en modalidad Promoción.
- Tener las entregas aprobadas sin haber re entregado ninguna de ellas.
- Deben tener aprobadas con final las materias correlativas antes de diciembre del corriente año.
- El promedio entre la nota de las entregas, coloquio y evaluación debe ser 6 o más.

Los alumnos que no cumplan con la promoción o estén inscriptos como regulares deberán entregar un trabajo final integrador para la aprobación final de la materia.



Bibliografía

- “An Introduction to Parallel Computing. Design and Analysis of Algorithms” Grama, Gupta, Karypis, Kumar, 2003.
- “Foundations of Multithreaded, Parallel and Distributed Programming” Andrews, 2000.
- “Introduction to HPC for Scientists and Engineers” Hager, Wellein, 2011.
- “An introduction to parallel programming” Peter Pacheco, 2011.
- “Parallel Programming for Multicore and Cluster Systems” Rauber, 2010.
- “Parallel programming in C with MPI and OpenMP” Quinn, 2003.
- “Sourcebook of Parallel Computing” Dongarra, Foster, Fox, 2003.

Cronograma tentativo

Cronograma de clases teóricas:

Clase	Contenidos/Actividades
1- Semana del 8/3	Unidad 1 (Conceptos básicos) – Unidad 2 (Arquitecturas Paralelas - Clasificación)
2- Semana del 15/3	Unidad 2 (Arquitecturas Paralelas - Evolución)
3- Semana del 22/3	Feriado (24 de marzo)
4- Semana del 29/3	Unidad 3 (Sistema de memoria)
5- Semana del 5/4	Unidad 4 (Principios de Diseño de algoritmos paralelos)
6- Semana del 12/4	Unidad 5 (Programación de algoritmos paralelos. Modelo de memoria compartida) Pthreads
7- Semana del 19/4	Unidad 5 (Programación de algoritmos paralelos. Modelo de memoria compartida) OpenMP
8- Semana del 26/4	Unidad 6 (Programación de algoritmos paralelos. Modelo de memoria distribuida) MPI
9- Semana del 3/5	Unidad 7 (Programación de algoritmos paralelos. Modelo híbrido)
10- Semana del 10/5	Unidad 8 (Métricas)
11- Semana del 17/5	Unidad 9 (Paradigmas)
12- Semana del 24/5	Unidad 10 (Introducción a la programación sobre GPUs)

Cronograma de prácticas experimentales:

Práctica	Contenidos/Actividades
1- Semana del 15/3	Práctica 1 – Optimización de algoritmos secuenciales
2- Semana del 22/3	Práctica 1 – Optimización de algoritmos secuenciales
3- Semana del 12/4	Práctica 2 – Modelo de memoria compartida - Pthreads
4- Semana del 19/4	Práctica 2 – Modelo de memoria compartida - Pthreads
5- Semana del 26/4	Práctica 3 – Modelo de memoria compartida - OpenMP
6- Semana del 3/5	Práctica 3 – Modelo de memoria compartida - OpenMP
7- Semana del 10/5	Práctica 4 – Modelo de memoria distribuida - MPI
8- Semana del 17/5	Práctica 4 – Modelo de memoria distribuida - MPI
9- Semana del 24/5	Práctica 5 – Híbridos e introducción a la programación sobre GPUs



**UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA**

Cronograma de entregas previstas y coloquios:

Fecha tentativa	Contenidos/Actividades
1- Semana del 5/4	Entrega 1
2- Semana del 26/4	Re entrega 1
3- Semana del 7/6	Entrega 2
4- Semana del 28/6	Re entrega 2
5- Semanas 29/6 al 16/7	Coloquios

Cronograma de evaluaciones previstas:

Fecha tentativa	Contenidos/Actividades
1- Semana del 14/6	Evaluación
2- Semana del 28/6	Primer recuperatorio
3- Semana del 12/7	Segundo recuperatorio

Contacto con la cátedra:

Plataforma virtual: <https://ideas.info.unlp.edu.ar>
Mail: apousa@lidi.info.unlp.edu.ar

Firmas del profesor responsable: