

PROGRAMA DE LA ASIGNATURA:
Sistemas Paralelos (IF061)

CÓDIGO: IF061
AÑO DE UBICACIÓN EN EL PLAN DE ESTUDIOS:
5 año
FECHA ULTIMA REVISIÓN DE LA ASIGNATURA:
2021-03-04
CARRERA/S: Licenciatura en Sistemas 049/2017,

CARÁCTER: CUATRIMESTRAL (1ro)
TIPO: OBLIGATORIA
NIVEL: GRADO
MODALIDAD DEL DICTADO: PRESENCIAL (EN LÍNEA)
MODALIDAD PROMOCION DIRECTA: NO
CARGA HORARIA SEMANAL: 6 HS
CARGA HORARIA TOTAL: 90 HS

EQUIPO DOCENTE

Nombre y Apellido	Cargo	e-mail
Enzo Rucci	Adjunto - contrato	erucci@lidi.unlp.edu.ar
Federico Gonzalez	Adjunto	fgonzalez@untdf.edu.ar

1. FUNDAMENTACION

Los conceptos de Concurrencia y Paralelismo son centrales en la Informática actual. El aprendizaje de los fundamentos, metodologías y herramientas de la programación paralela resultan fundamentales para la formación en cualquiera de las carreras de Informática. Desarrollar aptitudes para programar eficientemente aplicaciones paralelas (desde diferentes métricas, en particular tiempo de respuesta y consumo energético) resulta de gran valor para los futuros profesionales.

El temario de la asignatura se ajusta a las pautas de acreditación de las carreras de Informática en el país.

2. OBJETIVOS

a) 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.

b) OBJETIVOS ESPECIFICOS

Resolver problemas de programación paralela con diferentes modelos, sobre arquitecturas físicas

concretas (Cluster y Cloud).

Explicitar la importancia conceptual del paralelismo y su aplicación en todas las áreas de la Informática actual, valorizando los nuevos ejes de investigación de rendimiento, en particular el consumo energético.

3. CONDICIONES DE REGULARIDAD Y APROBACION DE LA ASIGNATURA

El régimen de regularidad se mantiene sin modificaciones mientras dure la pandemia por COVID-19. Para aprobar la cursada, los alumnos deben desarrollar 4 trabajos prácticos que involucran experimentación, los cuales deben defenderse en un coloquio breve en forma remota. Cada uno de los trabajos prácticos cuentan con 1 fecha recuperatoria.

Las condiciones de promoción sí se ven modificadas, ya que el examen teórico de promoción se tomará de forma oral en vez de escrita. Se mantiene la posibilidad de complementar el examen anterior con un trabajo monográfico de investigación que permita profundizar algunos de los temas vistos, en caso de ser necesario.

Se mantiene sin cambios la posibilidad de rendir examen final durante la duración de la validez de trabajos prácticos que podrá reemplazarse por un trabajo final integrador a desarrollar por el alumno.

4. CONTENIDOS DE LA ASIGNATURA

CONTENIDOS MÍNIMOS:

Algoritmos Paralelos
Paradigmas de resolución de sistemas paralelos
Métricas de performance
Arquitecturas de procesamiento paralelo.
Modelos de memoria compartido y/o lenguajes
Modelos y sistemas operativos para procesamiento paralelo
Optimización de algoritmos paralelos
Aplicaciones.

Unidad 1: Conceptos básicos
Paralelismo. Objetivos del procesamiento paralelo.
Proceso y Procesador. Interacción, comunicación y sincronización de procesos.
Concurrencia y Paralelismo. Modelos de Concurrencia.
Impacto del procesamiento paralelo sobre los sistemas operativos y lenguajes de programación.
Concepto de Sistema Paralelo.
Speedup y Eficiencia de algoritmos paralelos.
Concepto de asignación de tareas y balance de carga.
Balance de carga estático y dinámico.
Conceptos de Green Computing.

Unidad 2: Arquitecturas orientadas a Procesamiento Paralelo
Paralelismo implícito: tendencias en las arquitecturas de microprocesadores.
Optimización de performance en los sistemas de memoria. Manejo de memoria cache.
Estructura de control y modelos de comunicaciones en plataformas de procesamiento paralelo.
Conceptos de Multicores y GPGPUs.
Clasificación por mecanismo de control (SISD. SIMD. MISD. MIMD), por la organización del espacio de direcciones, por la granularidad de los procesadores y por la red de Interconexión.

Análisis del impacto del tiempo de comunicación en el speedup alcanzable.
Vector processors, array processors, Arquitecturas cúbicas e hipercúbicas. Supercomputadoras.
Clusters de PCs. Multiclusters. Grid. Cloud
Modelos de comunicación en Sistemas Paralelos.
Optimización de las comunicaciones en esquemas multiprocesador.

Unidad 3: Principios de diseño de algoritmos paralelos

Técnicas de descomposición.
Características de los procesos. Interacción.
Técnicas de mapeo de procesos/procesadores. Balance de carga.
Métodos para minimizar el overhead de la interacción entre procesos.
Modelos de algoritmos paralelos.
Problemas paralelizables y no paralelizables.
Paralelismo perfecto. Paralelismo de datos. Paralelismo de control. Paralelismo mixto.

Unidad 4: Modelos y Paradigmas de Computación Paralela

Parallel Random Access Machine (PRAM)
Bulk Synchronous Parallel (BSP)
LogP. Otras variantes de modelos analíticos.
Paradigma Master/Slave.
Paradigma Divide/Conquer.
Paradigma de Pipelining.
Metodología de diseño de algoritmos paralelos.

Unidad 5: Métricas del paralelismo

Medidas de performance standard.
Fuentes de overhead en procesamiento paralelo.
Speedup. Rango de valores. Speedup superlineal.
Eficiencia. Rango de valores. Grado de paralelismo alcanzable.
Efecto de la granularidad y el mapeo de datos sobre la performance.
Cargas de trabajo y modelos de speedup. Modelo de carga fija (Amdahl). Modelo de tiempo fijo (Gustafson). Modelo de memoria limitada (Sun y Ni).
Escalabilidad de sistemas paralelos.
Concepto de isoeficiencia. Función de isoeficiencia.
Métricas relacionadas con el consumo energético.

Unidad 6: Programación de algoritmos paralelos con Pasaje de Mensajes

Principios de la comunicación/sincronización por pasaje de mensajes.
Primitivas Send y Receive. La interfaz MPI como modelo.
Cómputo y Comunicaciones
Comunicaciones colectivas y operaciones de procesamiento.
Ejemplos sobre arquitecturas multiprocesador.

Unidad 7: Programación de algoritmos paralelos con memoria compartida.

Concepto de thread.
Primitivas de sincronización en PThreads.
Control de atributos en threads.
OpenMP como modelo Standard.
Análisis de problemas.

Unidad 8: Conceptos de Arquitecturas GRID y CLOUD.

Clusters, multiclusters y GRID.

Extensión a Cloud Computing.
 Extensión de conceptos de Cluster-processing a Grid y Cloud.
 Manejo de procesamiento sobre Cloud.

Unidad 9: Aplicaciones: Algoritmos paralelos clásicos.
 Presentación de casos clásicos:
 Sorting / Algoritmos sobre grafos /Procesamiento de matrices.
 Algoritmos de búsqueda para optimización discreta.
 Programación dinámica.
 Análisis de soluciones sobre diferentes arquitecturas paralelas.

5. RECURSOS NECESARIOS

6. PROGRAMACIÓN SEMANAL

Semana	Unidad / Módulo	Descripción	Bibliografía
Semana 1	Módulo I	Unidad 1: Conceptos básicos	
Semana 2	Módulo I	Unidad 1: Conceptos básicos	
Semana 3	Módulo II	Arquitecturas orientadas a Procesamiento Paralelo	
Semana 4	Módulo II	Arquitecturas orientadas a Procesamiento Paralelo	
Semana 5	Módulo III	Principios de diseño de algoritmos paralelos	
Semana 6	Módulo III	Principios de diseño de algoritmos paralelos	
Semana 7	Módulo IV	Modelos y Paradigmas de Computación Paralela	
Semana 8	Módulo IV	Modelos y Paradigmas de Computación Paralela	
Semana 9	Módulo V	Métricas del paralelismo	
Semana 10	Módulo V	Métricas del paralelismo	
Semana 11	Módulo VI	Programación de algoritmos paralelos con Pasaje de Mensajes	
Semana 12	Módulo VII	Programación de algoritmos paralelos con memoria compartida	
Semana 13	Módulo VIII	Conceptos de Arquitecturas GRID y CLOUD	
Semana 14	Modulo IX y entrega de la actividad	Aplicaciones: Algoritmos paralelos clásicos. Entrega de la actividad	
Semana 15	Consultas	Consultas previas a evaluación	
Semana 16	Evaluación	Evaluación	
Semana 17	Informe final de cátedra	Informe final de cátedra	

7. BIBLIOGRAFIA DE LA ASIGNATURA

Gregory R. Andrews 2000. Foundations of Multithreaded, Parallel and Distributed Programming. Addison Wesley.

Grama, Gupta, Karypis, Kumar 2003. Introduction to Parallel Computing. Addison Wesley.

Wilkinson, Allen 2005 Parallel Programming completo Complementaria Prentice Hall.

Dongarra, Foster, Fox, Gropp, Kennedy, Torczon, White 2003. Sourcebook of Parallel Computing. Morgan Kauffman.

Peter Pacheco 2011. An introduction to parallel programming. Elsevier.

Thomas Rauber, Gudulla Runger 2010. Parallel Programmng for Multicore and Cluster Systems. Springer.

Georg Hager, Gerard Wellein 2011. Introduction to High Performance Computing for Scientists and Engineers completo Complementaria CRC Press.

Henessy, Patterson 2012. Computer Architecture. A quantitative approach. 5ta. Ed. Elsevier.

Chapman, Jost, Van der Pas 2008. Using OpenMP. Portable Shared Memory Parallel Programming. MIT Press.

Firma del docente-investigador responsable

VISADO		
COORDINADOR DE LA CARRERA	DIRECTOR DEL INSTITUTO	SECRETARIO ACADEMICO UNTDF
Fecha :	Fecha :	