



SISTEMAS PARALELOS

Año 2015

Carrera/ Plan:

Licenciatura en Informática Plan
2003/2007/2012/2015
Licenciatura en Sistemas Plan
2003/2007/2012/2015

Año: 4°

Régimen de Cursada: Semestral

Carácter:

Obligatoria (LI)
Optativa (LS)

Correlativas: Programación Concurrente

Profesor: Franco Chichizola

Hs. semanales: 6 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 performance 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 MINIMOS:

- Arquitecturas de procesamiento paralelo.
- Modelos de comunicación. Métricas de performance.
- Memoria compartida, Memoria distribuida, esquemas mixtos.
- Lenguajes y sistemas operativos para procesamiento paralelo.
- Paradigmas de resolución de sistemas paralelos.
- Adaptación entre arquitectura y software.
- Aplicaciones.
- Arquitecturas de almacenamiento
- Green computing



PROGRAMA ANALÍTICO

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.
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.
Conceptos de procesadores dedicados a la aplicación, tipo DSP.
Clusters de PCs. Multiclusters. Grids.
Arquitecturas de almacenamiento.

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: Operaciones de Comunicación en arquitecturas multiprocesador

One-to-all Broadcast y All-to-one Reduction.
All-to-all Broadcast y Reduction.
Operaciones de All-Reduce y Prefix-Sum.
Comunicaciones de Scatter and Gather.
Comunicaciones All-to-All personalizadas.
Shift circular.
Optimización en las comunicaciones en esquemas multiprocesador.

Unidad 5: 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 6: 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 7: 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 8: Programación de algoritmos paralelos sobre plataformas con memoria compartida.

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

Unidad 9: Conceptos de Arquitecturas GRID.

Clusters, multiclusters y GRID.
Extensión de conceptos de Cluster-processing a Grid-processing.
Funcionalidades del Middleware en arquitecturas GRID.
Algoritmos sobre GRID.
Modelos y paradigmas de Sistemas Paralelos extendidos a GRID.
Análisis de casos de sistemas GRID.

Unidad 10: 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.



METODOLOGÍA DE ENSEÑANZA Modalidad presencial

La asignatura se estructura con clases teóricas, explicaciones de práctica y prácticas experimentales:

- Las clases teóricas son dictadas por el profesor de la asignatura y no son obligatorias.
- Las explicaciones de práctica son introductorias al trabajo en Laboratorio, para facilitar la utilización del equipamiento y software por los alumnos. Si bien no son obligatorias, es recomendado que los alumnos asistan.
- Las clases prácticas consisten en el desarrollo de trabajos con diferentes arquitecturas paralelas y lenguajes de programación. Los trabajos se pueden realizar individualmente o en grupo de 2 personas. Las consultas y correcciones son realizadas en forma presencial en los horarios de práctica definidos, o por medio de la plataforma de Educación a Distancia de la UNLP (WEBUNLP) en el resto de la semana.

Todos los trabajos son realizados en Laboratorio, sobre diferentes arquitecturas paralelas que dispone la Facultad.

EVALUACIÓN Modalidad presencial

Los alumnos deben aprobar las entregas de los diferentes trabajos experimentales a realizar y su posterior coloquio. Cada trabajo cuenta con más de una entrega donde se evalúa por parte de los auxiliares y se le explica a el/los alumno/s los errores que se deben corregir.

La aprobación de estos trabajos experimentales otorga la aprobación de la cursada de la asignatura.

La aprobación final de la materia puede ser mediante:

- examen teórico/práctico en la mesa de final en que se inscriba el alumno.
- examen conceptual (al terminar la cursada) más proyecto específico que el alumno resuelve, presenta una monografía sobre el tema y defiende en un coloquio en una fecha de examen final.

METODOLOGÍA DE ENSEÑANZA Modalidad no presencial

Se hace notar que por la característica de las tareas experimentales, el alumno deberá tener acceso a algún modelo de arquitectura multiprocesador configurado como cluster para poder realizar los trabajos que se solicitan en el curso.

Se buscará el modo de que utilicen equipamiento que tengan accesible en su ámbito laboral o darles un acceso remoto a determinadas facilidades en la Facultad para realizar los trabajos prácticos.

EVALUACIÓN Modalidad no presencial

Los alumnos en esta modalidad, deberán presentarse al menos 2 veces en el cuatrimestre para defender y aprobar las entregas de los diferentes trabajos experimentales que hayan realizado. En caso de no aprobar una entrega, tendrán una fecha general de recuperatorio.

Las fechas y horarios se establecerán en el calendario de la asignatura, una vez definido quienes se anotan en modalidad no presencial.

La aprobación de estos trabajos experimentales otorga la aprobación de la cursada de la asignatura.

La aprobación final de la materia puede ser mediante:

- examen teórico/práctico en la mesa de final en que se inscriba el alumno.
- examen conceptual (al terminar la cursada) más proyecto específico que el alumno resuelve, presenta una monografía sobre el tema y defiende en un coloquio en una fecha de examen final.



BIBLIOGRAFÍA OBLIGATORIA

- Introduction to Parallel Computing . Grama, Gupta, Karypis, Kumar. Addison Wesley 2003
- Foundations of Multithreaded, Parallel and Distributed Programming . Andrews. Addison Wesley 2000.
- Parallel Programming . Wilkinson, Allen. Prentice Hall 2005.
- Sourcebook of Parallel Computing . Dongarra, Foster, Fox, Gropp, Kennedy, Torczon, White. Morgan Kauffman 2003.
- Parallel and Distributed Computing. A Survey of Models, Paradigms and Approaches . Leopold. Wiley, 2001.

BIBLIOGRAFÍA COMPLEMENTARIA

- Parallel Computation. Models and Methods . Akl. Prentice Hall 1997.
- S C - @ - - 2 Parallell Algorithms . Akl. Prentice Hall, Inc. 1989.
- Studies in Computational Science. Parallel Programming Paradigms . Brinch Hansen. Prentice Hall, 1995.
- Parallel Program Design. A Foundation . Chandy, Misra. Addison Wesley, 1988.
- Distributed and Parallel Computing. El-Rewini H., Lewis T. Manning Publications, 1998.
- PVM: Parallel Virtual Machine - A Users Guide and Tutorial for Network Parallel Computing . Geist, Beguelin , Dongarra , Jiang, Mancheck , Sunderam. MIT Press, 1994.
- Communicating Sequential Processes . Hoare , Englewood Cliffs. Prentice Hall, 1985
- Computer Architecture and Parallel Processing . Hwang, Briggs. McGraw-Hill Inc, 1984
- Scalable Parallel Computing . Hwang, Xu. McGraw-Hill, 1998.
- Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes . Leighton. Morgan Kaufmann Publishers Inc, San Mateo, California, 1992
- Introduction to Parallel Computing . Lewis, El-Rewini. Prentice-Hall, Inc., Englewood Cliffs, 1992.
- In Search of Clusters . Pfister. Prentice Hall, 2nd Edition, 1998.
- MPI: The complete Reference . Snir, Otto, Huss-Lederman, Walker, Dongarra, Cambridge. MIT Press, 1996.
- Procesamiento Paralelo. Conceptos de Arquitecturas y Algoritmos . Tinetti, De Giusti. Editorial Exacta 1998.
- GRID Computing: A practical guide to technology and applications . Abbas. Charles River Media 2004.
- The GRID 2. Blueprint for a new computing infrastructure . Foster, Kesselman. Morgan Kauffman 2004.
- IEEE, ACM Digital Library



CRONOGRAMA TENTATIVO DE CLASES Y EVALUACIONES

El comienzo de clases está previsto para la semana del 24 de marzo.

El cronograma detallado se pone en conocimiento de los alumnos al inicio del curso.

El esquema de las clases teóricas es el siguiente:

Clase	Contenidos/Actividades	Evaluaciones previstas
1- Semana del 23/03	Introducción al Procesamiento Paralelo. Conceptos. Fundamentos Sistemas Paralelos. Aplicaciones. Sistemas Distribuidos y Paralelos.	
2- Semana del 30/03	Plataformas de procesamiento para Sistemas Paralelos. Evolución de las arquitecturas y las comunicaciones. Relación con los Sistemas Distribuidos.	
3- Semana del 06/04	Conceptos de programación distribuida y paralela. Aplicaciones de los sistemas paralelos. Programación en Memoria Compartida. Modelos de programación basados en directivas OpenMP.	
4- Semana del 13/04	Programación en Memoria Compartida. Modelos basados en threads (Pthreads). Programación en Memoria Compartida y Rendezvous (ADA).	
5- Semana del 20/04	Programación de Algoritmos Paralelos utilizando el paradigma de Pasaje de Mensajes. MPI. PVM.	
6- Semana del 27/04	Modelos analíticos de programas paralelos. Métricas de Performance. Granularidad. Escalabilidad. Isoeficiencia. Grado de concurrencia.	
7- Semana del 04/05	Principios de diseño de Algoritmos Paralelos. Descomposición en tareas, granularidad de aplicaciones, mapeo de tareas.	
8- Semana del 11/05	Técnicas de descomposición de aplicaciones. Características de las tareas y las interacciones generadas.	
9- Semana del 18/05	Mapping de tareas a procesadores. Algoritmos de mapping para modelos basados en grafos. Técnicas de mapping para lograr balance de carga en modelos basados en descomposición de datos.	
10- Semana del 25/05	Introducción a GRID. Grid Computing y Cloud Computing. GPUs. Consumo Energético.	



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

El esquema de las clases prácticas es el siguiente:

Clase	Contenidos/Actividades	Evaluaciones previstas
Semana del 06/04 y 13/4	Optimización de algoritmos secuenciales	
Semana del 20/04	Practica OpenMP	
Semana del 27/04	Practica Pthreads	
Semana del 04/05	Practica ADA	
Semana del 11/05 y 18/05	Practica MPI	
Semana del 25/05	Presentación de trabajos a realizar para la cursada	
Semana del 01/06 hasta 22/06	Consultas sobre el trabajo a realizar.	
Semana del 29/06		Entrega/coloquio del trabajo de la cursada
Semana del 06/07	Consultas sobre las correcciones del trabajo.	
Semana del 13/07		Reentrega/coloquio del trabajo de la cursada

Aclaración respecto a la evaluación de la cursada: las entregas/coloquios de los trabajos con los cuales se aprueba la cursada se harán durante el horario de práctica de la semana indicada en el cronograma. Se debe tener en cuenta que los alumnos pueden realizar las entregas al terminar el trabajo (sin necesidad de haber llegado a la semana que indica el cronograma).

Contacto de la cátedra (mail, página, plataforma virtual de gestión de cursos):

Plataforma virtual: webunlp.unlp.edu.ar

Web: <http://weblidi.info.unlp.edu.ar/catedras/paralela/>

Mail: sparalelo@lidi.info.unlp.edu.ar

Firmas del/los profesores responsables: