Especialización en Cómputo de Altas Prestaciones – Modalidad a distancia

Programación Paralela sobre Arquitecturas Multiprocesador

Año 2021

Duración: 70 hs. Totales.

Cantidad de horas presenciales/VC: 20 hs.

Cantidad de horas de actividades en línea y de trabajo final: 50 hs.

OBJETIVOS GENERALES:

Caracterizar los problemas de procesamiento paralelo desde el punto de vista del sotware: descomposición en procesos, mapeo de procesos a procesadores, lenguajes de programación, mecanismos de empleo de bibliotecas para incorporar a lenguajes tradicionales, comunicación y sincronización por memoria compartida y por mensajes y transformación de algoritmos secuenciales en paralelos.

Describir los modelos y paradigmas de programación paralela, ejemplificando con problemas clásicos (numéricos y no numéricos).
Definir y discutir las métricas de performance asociadas al paralelismo, considerando el rendimiento y también requerimientos de eficiencia de importancia actual como el consumo energético.

Plantear casos concretos de procesamiento paralelo, resolubles sobre distintas arquitecturas multiprocesador, en particular multicores y clusters de multicores.

COMPETENCIAS A DESARROLLAR EN RELACION CON EL OBJETIVO DE LA CARRERA

C.1- Analizar problemas del mundo real que por su complejidad y/o volumen de datos requieran cómputo paralelo y diseñar soluciones desde el punto de vista del hardware necesario, lo que requiere un conocimiento de las arquitecturas paralelas actuales.

C.2- Conocer los fundamentos para el desarrollo de Sistemas Paralelos (incluyendo la relación entre hardware y software).

C.3- Tener capacidad de análisis, diseño, implementación y optimización de algoritmos distribuidos y paralelos, aplicables a problemas numéricos y no numéricos en diferentes áreas del conocimiento, incluyendo su análisis de rendimiento y eficiencia.

C.7- Estar capacitado para el desarrollo de aplicaciones paralelas sobre diferentes arquitecturas.

CONTENIDOS MINIMOS:

- Lenguajes para Concurrencia y Paralelismo. Expresión en los lenguajes de programación de aplicaciones.
- Mecanismos de Comunicación y Sincronización entre procesos y el modo en los lenguajes manejan estos mecanismos.
- Del enunciado de un problema "paralelizable" a su programación: descomposición y mapeo de procesos y procesadores.
- Modelos y paradigmas para la programación de Algoritmos Paralelos. Casos de estudio con lenguajes y bibliotecas específicas.
- Aplicación de métricas del paralelismo a programas paralelos específicos.
- Programación de algoritmos paralelos sobre arquitecturas de memoria compartida / distribuida e híbridas.
- Programación de algoritmos paralelos sobre arquitecturas multicore y clusters. Aplicaciones numéricas y no numéricas.

PROGRAMA

Conceptos básicos

Paralelismo. Procesos y Procesadores. Interacción, comunicación y sincronización de procesos. Lenguajes orientados a la programación paralela.
Expresión de la comunicación y sincronización entre procesos paralelos en diferentes lenguajes.
Speedup y Eficiencia de algoritmos paralelos.
Concepto de asignación de tareas y balance de carga. Balance de carga estático y dinámico.
Escalabilidad. Relación con la eficiencia.
Problemas paralelizables y no paralelizables.
Análisis de casos clásicos de programación paralela.

Diseño e Implementación de algoritmos paralelos. Modelos y Paradigmas.

Técnicas de descomposición. Características de los procesos. Interacción.
Técnicas de mapeo de procesos/procesadores. Balance de carga.
Modelos de algoritmos paralelos. Metodología de diseño de algoritmos paralelos.
Paralelismo de datos. Paralelismo de control.
Estudio de casos de paralelización de problemas numéricos y no numéricos.
Superlinealidad: concepto y casos concretos de algoritmos paralelos superlineales.

Programación de algoritmos paralelos con Pasaje de Mensajes
Comunicación/sincronización por pasaje de mensajes. Biblioteca MPI.
Implementación en diferentes lenguajes vinculados con la biblioteca MPI.
Estudio de diferentes lenguajes combinados con MPI. Ejemplos.
Resolución de algoritmos paralelos sobre procesadores multicores utilizando pasaje de mensajes. Evaluación de eficiencia y escalabilidad.
Resolución de algoritmos paralelos sobre clusters con multicores utilizando pasaje de mensajes. Evaluación de eficiencia y escalabilidad.
Cómputo y Comunicaciones en los casos estudiados. Relación con la eficiencia.
Estudio de métricas y consumo en casos concretos.

Programación de algoritmos paralelos sobre plataformas con memoria compartida.
Concepto de thread.
Primitivas de sincronización en PThreads. Combinación de lenguajes con PThreads.
Control de atributos en threads.
OpenMP como modelo Standard.
Resolución de algoritmos paralelos sobre procesadores multicores utilizando memoria compartida. Evaluación de eficiencia y escalabilidad.
Resolución de algoritmos paralelos sobre clusters con multicores utilizando memoria compartida. Evaluación de eficiencia y escalabilidad.
Análisis de casos en que se combina pasaje de mensajes y memoria compartida.
Estudio comparativo de soluciones paralelas con pasaje de mensajes y con memoria compartida.
Estudio de métricas y consumo en casos concretos.

Introducción a la programación paralela sobre otras arquitecturas multiprocesador.
Conceptos de programación sobre GPUs.
Conceptos de programación sobre placas aceleradoras específicas.
Conceptos de programación sobre Cloud.
Relación con los fundameentos de la programación paralela vistos en el curso.

ACTIVIDADES EXPERIMENTALES y DE INVESTIGACION

Tareas en Laboratorio (presencial o remoto)

Tal como se explica en el item relacionado con la metodología, ésta se basa en clases sincrónicas (presenciales o remotas) combinadas con actividades demostrativas en el laboratorio para aplicar los conceptos teóricos y que así el alumno adquiera las competencias y habilidades sobre cada uno de los temas que forman parte del contenido de la asignatura.

Además el alumno debe analizar un proyecto/desarrollo relacionado con los temas dictados en la teoría, cuya implementación concreta se realiza sobre máquinas / placas específicas y/o una infraestructura virtualizada que los alumnos pueden acceder en forma remota (en el Laboratorio dedicado a Paralelismo en el Postgrado).

Investigación/ Estudios adicionales:

Los alumnos analizarán papers relacionados con los problemas de paralelización de algoritmos y el proceso de análisis / diseño e implementación de los mismos, en particular sobre arquitecturas basadas en procesadores multicore.

Se les propondrán temas de I+D orientados al estudio comparativo de métricas de rendimiento en algoritmos sobre sistemas paralelos, de modo de potenciar el conocimiento trasmitido en la teoría.

METODOLOGIA Y MODALIDAD DE EVALUACION

La metodología se basa en clases sincrónicas a través del sistema de videoconferencias adoptado por el Postgrado de Informática combinadas con sesiones en el laboratorio remoto para aplicar los conceptos teóricos y que así el alumno adquiera las competencias y habilidades sobre cada uno de los temas que forman parte del contenido de la asignatura.

Se requiere un 80% de asistencia a los encuentros sincrónicos, incluyendo el encuentro inicial de presentación de la materia, y el encuentro final de integración, ambos de asistencia obligatoria.

El trabajo se complementa con un proyecto experimental que debe desarrollar el alumno para cumplimentar las horas asignadas con soporte tutorizado por el profesor (*on-line*) y seguimiento a través del Entorno Virtual IDEAS contemplado en el SIED de la Facultad de Informática de la UNLP.

El despliegue práctico se realizará sobre una infraestructura virtualizada accesible al alumno en la que se puede ejecutar aplicaciones y medir su rendimiento con las técnicas explicadas en el curso.
La evaluación se realizará mediante un examen escrito al final de las sesiones sincrónicas para evaluar el grado de conocimientos del alumno (20%), el proyecto/desarrollo experimental que deberá entregar el alumno al final de las horas programadas (70%) y la participación y aportaciones de calidad/excelencia a las soluciones propuestas (10%).

RECURSOS Y MATERIALES DE ESTUDIO

Como materiales de estudio, se dispone de:

- Presentaciones multimedia desarrolladas ad-hoc para introducir cada uno de los diferentes ejes temáticos.
- Píldoras formativas con la explicación de algunos temas
- Ejemplos donde se aplican los conceptos teóricos
- Ejercicios prácticos que son desarrollados en clase
- Material de lectura para estudiar y profundizar conceptos abordados en las clases
- Enlaces a artículos de actualidad de repositorios reconocidos en el área
- Acceso a equipamiento remoto situado en la Facultad de Informática de la UNLP y también en la nube (Cloud) de acuerdo a la disponibilidad del Postgrado de la Facultad de Informática para sus cursos.

- Software específico para determinadas actividades de laboratorio que se detallan en este programa.

ACTIVIDADES EXPERIMENTALES Y APROPIACIÓN DE SABERES

Los trabajos experimentales pueden desarrollarse en cada clase o continuarse en más de una clase. Parten de una especificación/consigna del docente (explicada en la clase) y un trabajo individual o en grupos que interactúan en el que los alumnos resuelven un problema experimental concreto relacionado con la temática.

Los trabajos podrán ser individuales o grupales. Para esto último se configura el entorno virtual para que los alumnos del mismo grupo se encuentren en un espacio virtual diferente del resto. Durante el desarrollo del trabajo, el docente estará conectado respondiendo dudas y consultas.

Estos trabajos pretenden desarrollar y/o fortalecer las aptitudes de opinión crítica en los temas relativos del curso. Los alumnos deberán sintetizar su comprensión de los temas, al realizar correctamente la tarea experimental propuesta.

También se pretende desarrollar la capacidad de poder comunicar y transmitir los resultados, en presentaciones pautadas a lo largo del curso.

En general, finalizada una actividad, hay una sesión de discusión conjunta donde los participantes comunicarán sus opiniones e intercambiarán los distintos puntos de vista.

BIBLIOGRAFÍA BASICA

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.

MPI: The complete Reference
Snir, Otto, Huss-Lederman , Walker , Dongarra , Cambridge, MA: MIT Press, 1996.

Introduction to High Performance Computing for Scientists and Engineers".
Georg Hager, Gerard Wellein. CRC Press (2011).

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

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

Parallel Programming in OpenMP.
Chandra, Dagum, Kohr, Maydan, McDonald,Menon. Morgan Kauffman (2011).

BIBLIOGRAFÍA COMPLEMENTARIA

Cloud Computing: A Hands-On Approach
Arshdeep Bahga, Vijay Madisetti. CreateSpace Independent Publishing Platform, 2013. ISBN-13: 978-1494435141

Programming Massively Parallel Processors: A Hands-on Approach (Applications of GPU Computing Series)
David B. Kirk, Wen-mei W. Hwu. Morgan Kaufmann, 2010. ISBN-13: 978-0123814722

CUDA Programming: A Developer's Guide to Parallel Computing with GPUs (Applications of Gpu Computing)
Shane Cook. Morgan Kaufmann, 2012. ISBN-13: 978-0124159334

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

Programming Massively Parallel Processors".
Kirk, Hwu. Elsevier 2010.

Computer Architecture. A quantitative approach.
Henessy, Patterson.5ta edición. Elsevier (2012).

Using OpenMP. Poratble Shared Memory Parallell Programming.
Chapman, Jost, Van der Pas.MIT Press (2008).

Using MPI-2.
Gropp, Lusk, Thakur. The MIT Press (1999).

IEEE, ACM Digital Library