Nombre de la Asignatura: Procesamiento en Paralelo
Línea de Investigación o Trabajo: Automática e Informática Industrial
Tiempo de dedicación del estudiante a las actividades de:
DOC-TIS-TPS-CRÉDITOS
48 60 0 6

DOC: Docencia; TIS Trabajo Individual Significativo; TPS Trabajo Profesional Supervisado

1. Historial de la Asignatura. Establece información referente a lugar y fecha de elaboración y revisión, quiénes participaron en su definición y algunas observaciones académicas.

Fecha revisión / actualización	Participantes	Observaciones, cambios o justificación
Febrero 22 del 2012	Carmen Leticia García Mata	

2. **Pre-requisitos y corequisitos**.

Programación en C, Métodos Numéricos, Algebra Lineal. Deseable Algoritmos, Estructuras de Datos

3. **Objetivo de la asignatura.**

Dar una introducción al procesamiento en paralelo a fin de que el estudiante conozca la tendencias en arquitecturas paralelas, técnicas de programación y tipos de problemas que por su extensión ó necesidad de cómputo intensivo sea conveniente el uso de cómputo en paralelo para una solución más eficiente y veloz. Al término del curso el alumno deberá ser capaz de resolver problemas de cómputo intensivo en menos tiempo mediante el diseño de algoritmos paralelos e implementar estos algoritmos usando MPI, OpenMP ó una combinación de ambos. Adicionalmente, el estudiante deberá ser capaz de determinar los cuellos de botella computacionales y optimizar la eficiencia del código. También deberá ser capaz describir las diferentes arquitecturas en paralelo, las redes de interconexión, los modelos de programación y los algoritmos para las operaciones más comúnes.

4. Aportación al perfil del graduado.

Este curso contribuirá a que el graduado maneje con destreza herramientas avanzadas del área de computación para la solución computacional de problemas complejos tales como lenguajes, sistemas, librerías y técnicas avanzadas de computación, lo que le permitirá escoger la metodología más adecuada para resolver de manera más eficiente problemas de investigación de su área. Un aspecto importante del curso es la introducción del estudiante a cuestiones de análisis de algoritmos y teoría de complejidad lo que le permitirá identificar si los problemas son solubles o no y si lo son, cuál es el mejor tiempo que puede obtener en base a un algoritmo dado.

Podrá Identificar, evaluar y resolver problemas de su área que requieran de cómputo intensivo, ya sea por la complejidad del algoritmo ó por el volumen de datos que maneje el problema, mediante procesamiento en paralelo. El contenido temático le permitirá conocer las tendencias actuales en el área de cómputo de alto desempeño, lo que le permitirá dará un plus a su investigación. Aunque la paralelización de los sistemas de cómputo cada día se incrementa, no existe aún suficientes investigadores preparados para resolver este tipo de problemas y es una área altamente demandada tanto en investigación como múltiples empresas, por lo que el estudiante tendrá además una ventaja profesional importante cuando curse esta materia. Al término del curso el esutdiante podrá ser capaz de establecer cuál es la arquitectura, metodología, y sistemas para resolver un problema dado mediante cómputo paralelo.

5. Contenido temático.

Unidad	Temas	Subtemas
1. Introducción a Procesamiento en Paralelo	1.1 Problemas que requieren paralelización	3. Requerimentos de Problemas complejos: Velocidad vs. Memoria 4. Arquitecturas Paralelas: Breve introducción
	1.2 Modelos de Cómputo	5. Definiciones básicas 6. Evolución de las Computadoras Secuenciales 7. Tendencias Actuales en Diseño de Computadoras
2. Fundamentos de Arquitecturas Paralelas	2.1 Topologías de Interconexión	8. Memoria Distribuida vs. Memoria Compartida 9. Sistemas Comerciales 10. Clasificación de Flynn 11. Coherencia de Caché
	2.2 Mecanismos de Sincronización	12. Modelos de Memoria y Sincronización 13. Memoría Caché 14. Protocolo Basado en Directorio
	2.3 Introducción a Arquitecturas Paralelas	15. MIMD-Memoria Distribuida 16. Redes de Interconexión 17. Métodos de Comunicación
3. Cuestiones de Evaluación de Eficiencia	3.1 Evaluación de Eficiencia	18. Ley de Grosh 19. Speedup 20. Eficiencia 21. Ley de Amdhal 22. Ley de Gustafson-Barsis 23. Modelos de Eficiencia: Escalabilidad y Benchmarks 24. Balanceo de Carga, Granularidad, E/S
4. Programación Paralela Asíncrona	4.1 El Paradigma de Paso de Mensajes	25. Fundamentos 26. Ejemplos Históricos 27. Mensajes Síncronos 28. Comunicación

		Colectiva
	4.2 Modelos de Comunicación para Paso de Mensajes	29. Comunicación Punto a Punto 30. Comunicación Colectiva
	4.3 MPI-Biblioteca para Paso de Mensajes	31. Llamadas a Funciones Básicas 32. Reduce 33. Funciones de Sincronización 34. Barreras 35. Funciones para Comunicación Punto a Punto, Comunicación con Bloqueo y No Bloqueo y Comunicación Colectiva 36. Ejemplos con MPI 37. Técnicas de Evaluación y Depuración
		38.
5. Programación con Memoria Compartida Distribuida	5.1 Multiprocesadores y Sistemas con Memoria Compartida	39. Ventajas y Desventajas 40. Lenguajes, Sistemas y Librerías para Programación en Paralelo 41. Hilos vs. Procesos 42. Creación de Procesos, Tareas e Hilos den Unix 43. Modelo de los Hilos 44. Creación de Datos Compartidos con Hilos y Procesos
	5.2 Funcionalidades Requeridas	45. Memoria Segmentada: Funcionalidades y Creación de Procesos 46. Memoria Virtual: Mapeos y Creación de Procesos 47. Mecanismos de Sincronización: Mutexes, Semáforos, Monitores, Variables Condicionales 48. Deadlock 49. Sincronización de Procesos

6. Técnicas de Programación en Paralelo	6.1 Cómputo Paralelo Simple	50. Método Maestro-Esclavo 51. Método Work Pool
	6.2 Estrategias de Particionamiento Divide y Vencerás	52. Conceptos básicos 53. Construcción de Árboles 54. Ejemplos de Problemas Divide y Vence: Ordenamiento de Cubetas, Integración Numérica
	6.3 Embarrasing Parallelism	55. Computación del Conjunto Mandelbrot 56. Métodos Monte Carlo 57. Generación de Números Paralelos Aleatorios
	6.4 Particionamiento y Divide y Conquista	58. Particionamiento Estático 59. Método Divide y Conquista 60. Algoritmo para Ordenamiento de Cubetas
	6.5 Balanceo de Carga y Detección de Terminación	61. Balanceo de Carga Estática 62. Balanaceo de Carga Dinámica 63. Balanceo de Carga Centralizada 64. Balanceo de Carga Descentralizada 65. Detección de Terminación Distribuida
	6.6 Computación Síncrona	66. Conceptos Básicos 67. Ejemplos: Ecuaciones Lineales, El Juego de la Vida
7. Programación Paralela en Memoria Compartida con Open MP	7.1 Datos Compartidos	68. IPC, Comunicación entre Procesos 69. Memoria Virtual 70. Mecanismos de Sincronización
	7.2 OpenMP	71. Variables Condicionales 72. Candados entre Procesos 73. Introducción

		74. Conditional variables; locks between processes; introduction to OpenMP

6. **Metodología de desarrollo del curso.**

Se trabajará mediante presentaciones y discusiones del material tanto básico, proporcionado por el instructor como material expandido a través de búsquedas en la Web y Bibliográficas. Se plantearán experimentos de las ideas discutidas en clase en el cluster del Laboratorio de Sistemas Inteligentes y Visión por Computadora.

- **Sugerencias de evaluación.**

Se asignarán tareas de forma periódica consistentes en la implementación de algoritmos paralelos bajo diferentes metodologías . Se aplicarán pruebas cortas (quizzes) al final de cada unidad. Como proyecto final cada estudiante deberá escribir un artículo sobre un conjunto de temas que proporcione el instructor. El artículo deberá ser ó un análisis del estado del arte (survey) sobre un tema poco explorado o bien usar una herramienta para programación en paralelo y construir un tutorial para ésta.

- **Bibliografía y Software de apoyo**.

Levesque J. & Wagenbreth G. High Performance Computing: Programming and Applications (Chapman & Hall/CRC Computational Science)

T.G. Lewis and H. El-Rewini. Introduction to Parallel Computing, Prentice Hall, 1992.

B. Wilkinson and M. Allen. Parallel Programming, Prentice Hall, 1999.

I. Foster. Designing and Building Parallel Programs, Addison-Wesley, 1995.

M.J. Quin. Parallel Programming in C with MPI and OpenMP', McGraw Hill, 2004

G. Hager and G. Wellein: *Introduction to High Performance Computing for Scientists and Engineers.* CRC Press, ISBN 978-1439811924, 356 pages, July 2010.

G. Hager, G. Schubert, T. Schoenemeyer, and G. Wellein: *Prospects for Truly Asynchronous Communication with Pure MPI and Hybrid MPI/OpenMP on Current Supercomputing Platforms*. Accepted for the Cray Users Group Conference 2011 (CUG 2011), May 23-26, 2011, Fairbanks, AK. Hager-Paper-CUG11.pdf

9. Prácticas propuestas.

Unidad	Prácticas
1 y 2	Aprender a compilar y a ejecutar procesos en el cluster con diferentes opciones. Desarrollar e implementar algoritmos secuenciales para procesamiento de matrices. La ejecución de los algoritmos deberán mostrar tiempos de ejecución para diferentes configuraciones de los nodos. El objetivo es que el alumno estudie, identifique y aprenda cuáles son las llamadas a las funciones del sistema operativo Linux para realizar esta tarea.

3	Computar los límites teóricos de speedup usando las leyes de Gustafson-Barsis para un cierto problema. Computar cuál es el número mínimo de procesadores necesarios para que un programa en paralelo sea acelerado en un valor dado.
4	Escribir programas en MPI para medir el tiempo de envío de mensajes entre procesos bajo el esquema maestro-esclavo, para medir los tiempos de ejecución entre el envío de mensajes a los proceos de forma individual y mediante la operación de broadcasting. Escribir un programa para una operación elemental de matrices mediante cómputo paralelo con MPI
5 y 6	Implementar la solución al problema de procesamiento de imágenes para operaciones de bajo nivel, mediante dos métodos. En el más simple el maestro envía a los esclavos el número de renglones a cada procesador y los esclaves generan un mensaje para cada pixel. En el más avanzado, el código se deberá optimizar de forma que cada esclavo determine el número de renglones por sí mismo y regresar el resultado en un solo mensaje y no pixel por pixel
7	Resolver problemas implementando la programación de memoria compartida, usando fork para crear procesos, mecanismos de sincronización y la creación del segmento de memoria compartida para la ejecución

8. Nombre y firma del catedrático responsable

CARMEN LETICIA GARCÍA MATA