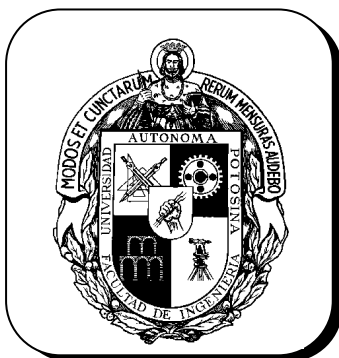


FACULTAD DE INGENIERÍA

ÁREA DE COMPUTACIÓN E INFORMÁTICA



Nombre de la materia: SUPERCÓMPUTO
Clave de la materia: 2226
Clave Facultad:
Clave U.A.S.L.P.: **Clave CACEI:** IA
Nivel del Plan de Estudios: II: 9, IC: 10 **No. de créditos:** 8
Horas/Clase/Semana: 3 **Horas totales/Semestre:** 80
Horas/Práctica (y/o Laboratorio): 2
Prácticas complementarias:
Trabajo extra-clase Horas/Semana: 3
Carrera/Tipo de materia: Común del Área/Optativa
No. de créditos aprobados:
Fecha última de Revisión Curricular: 12/03/2010
Materia y clave de la materia requisito:

PROPÓSITO DEL CURSO

El supercómputo es una de las tecnologías informáticas más avanzadas que han permitido abrir nuevos campos de estudio e investigación en diferentes ramas. Permite procesar enormes cantidades de información y realizar

billones de cálculos matemáticos. Es necesario, por lo tanto, estudiar la arquitectura y el desarrollo de algoritmos que permitan utilizar al máximo dicha herramienta.

OBJETIVO DEL CURSO

Analizar y conocerá los conceptos que definen al súper cómputo. Será capaz de identificar problemas que pueden ser resueltos con dicha tecnología. Desarrollar algoritmos

utilizando las técnicas de programación paralela, mediante el uso de diferentes lenguajes de programación que soportan esta tecnología.

CONTENIDO TEMÁTICO

1. Introducción al cómputo distribuido

Tiempo estimado: 5 horas

Objetivo:

Conocer los conceptos básicos del cómputo distribuido, así como sus principales características.

- 1.1 Introducción
- 1.2 Escalabilidad
- 1.3 Sincronización
- 1.4 Coordinación

2. Introducción a la programación paralela

Tiempo estimado: 10 horas

Objetivo:

Revisar los conceptos básicos del cómputo paralelo.

- 2.1 Ideas preliminares
- 2.2 Definiciones básicas
- 2.3 Paralelismo de control
- 2.4 Escalabilidad
- 2.5 Análisis
- 2.6 Ley de Amdhal

3. El modelo PRAM (Parallel Random Access Machine)

Tiempo estimado: 10 horas

Objetivo:

Estudiar algunos modelos teóricos de programación.

- 3.1 Ideas preliminares

3.2 El modelo RAM

3.3 El modelo PRAM

3.4 Algoritmos PRAM

3.5 Problemas clásicos

4. Arquitecturas paralelas

Tiempo estimado: 5 horas

Objetivo:

Conocer e implementar algunas de las características más importantes de las arquitecturas paralelas.

- 4.1 Introducción
- 4.2 Organización de procesadores
- 4.3 Topologías de redes de procesadores
- 4.4 Arquitecturas paralelas
- 4.5 Speed-up, eficiencia y paralelicibilidad

5. Lenguaje paralelo MPI

Tiempo estimado: 15 horas

Objetivo:

Aprender a programar en el lenguaje paralelo MPI.

- 5.1 Introducción a MPI
- 5.2 Comunicación punto a punto
- 5.3 Modelos y modos de comunicación
- 5.4 Comunicación básica
- 5.5 Comunicación con buffer
- 5.6 Recepción

- 5.7 Tipos de datos
- 5.8 Etiquetas y comunicadores
- 5.9 Operaciones colectivas
- 5.10 Barreras y broadcast
- 5.11 Recolección
- 5.12 Distribución
- 5.13 Comunicación todos con todos
- 5.14 Reducción
- 5.15 Modularidad
- 5.16 Tipos de datos derivados

6. Mapeo de procesos en procesadores y planificación

Tiempo estimado: 15 horas

Objetivo: Realizar una revisión de los métodos y algoritmos usados para mapear los procesos originados en algoritmos paralelos sobre algunas arquitecturas paralelas clásicas.

- 6.1 Introducción
- 6.2 Anillo en una malla 2-D
- 6.3 Malla 2-D en malla 2-D
- 6.4 Árbol binario completo en malla 2-D
- 6.5 Árbol binomial en malla 2-D
- 6.6 Grafos en hipercubos
- 6.7 Anillos y mallas en hipercubos

- 6.8 Balance de carga
- 6.9 Planificación estática

7. Algoritmos paralelos elementales

Tiempo estimado: 10 horas

Objetivo: Conocer e implementar algunos de los más importantes algoritmos paralelos.

- 7.1 Introducción
- 7.2 Clasificación de algoritmos MIMD
- 7.3 Algoritmos de reducción
- 7.4 Algoritmos de broadcast

8. Programación en paralelo con GPU

Tiempo estimado: 10 horas

Objetivo: El estudiante conocerá algunos de los lenguajes y herramientas utilizadas actualmente para la programación en paralelo con GPU.

- 8.1 Introducción
- 8.2 GPGPU (General-Purpose Computing on Graphics Processing Units)
- 8.3 Programación con GPU

METODOLOGÍA

Se realizarán exposiciones por parte del profesor y del alumno. Se realizarán prácticas para la implementación

de los algoritmos en el laboratorio y como tareas.

EVALUACIÓN

Los exámenes representan el 100% de la evaluación. Se realizarán tres exámenes parciales en las fechas establecidas por la facultad, de acuerdo al Reglamento de Exámenes. Para presentar el tercer examen parcial, el examen extraordinario, el examen a título y

regularización; es requisito hacer entrega del sistema desarrollado (proyecto) en la materia de acuerdo con los lineamientos establecidos en el Sistema de Proyectos de Desarrollo Tecnológico Integrador del área.

BIBLIOGRAFÍA

Bibliografía Básica

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

Pacheco P., *Parallel Programming with MPI*, Morgan Kaufmann Publishers, 1997

Gropp W., Lusk E., Skjellum A., *Using MPI - Portable Parallel Programming with the Message Passing Interface*, 2nd Edition, The MIT Press, 1999

Tanenbaum, A., *Sistemas operativos distribuidos*, Prentice-Hall, 1996

Bibliografía Complementaria

Coulouris G., Dollimore J., Kindberg T., *Distributed Systems - Concepts and Design*, Fourth Edition, Addison-Wesley, June 2005

Lazou, C., *Supercomputers and their use*, Oxford Clarendon Press, 1988

Chandra, R., *Parallel programming in OpenMP*, Morgan Kaufmann Publishers, 2001