

Universidad Católica San Pablo
Facultad de Ingeniería y Computación
Escuela Profesional de
Ciencia de la Computación
SILABO



CS314. Algoritmos Paralelos (Obligatorio)

2010-1

1. DATOS GENERALES

1.1 CARRERA PROFESIONAL	:	Ciencia de la Computación
1.2 ASIGNATURA	:	CS314. Algoritmos Paralelos
1.3 SEMESTRE ACADÉMICO	:	7 ^{mo} Semestre.
1.4 PREREQUISITO(S)	:	CS210T. Análisis y Diseño de Algoritmos. (5 ^{to} Sem) , CS225T. Sistemas Operativos. (6 ^{to} Sem)
1.5 CARÁCTER	:	Obligatorio
1.6 HORAS	:	2 HT; 2 HP; 2 HL;
1.7 CRÉDITOS	:	4

2. DOCENTE

3. FUNDAMENTACIÓN DEL CURSO

Las arquitecturas de computadores están tendiendo a incluir cada vez más núcleos y/o procesadores por máquina como método de incrementar la capacidad computacional de cada unidad. La posibilidad de realizar múltiples tareas simultáneamente mediante hardware no es inmediatamente traducida al software, pues las aplicaciones deben ser diseñadas para aprovechar estas nuevas capacidades, mediante el uso de hebras y/o procesos.

4. SUMILLA

1. CN/Computación Paralela.2. AR/Multiprocesamiento.3. AL/Algoritmos Paralelos.4. Modelos de Threads con PTHREADS 5. Modelos de Threads con OpenMP 6. Modelo de programación mediante paso de Mensajes con MPI 7. *Threading Building Blocks (TBB)*

5. OBJETIVO GENERAL

- Que el alumno sea capaz de crear aplicaciones paralelas de mediana complejidad aprovechando eficientemente máquinas con múltiples núcleos.
- Que el alumno sea capaz de comparar aplicaciones secuenciales y paralelas.
- Que el alumno sea capaz de convertir, cuando la situación lo amerite, aplicaciones secuenciales a paralelas de forma eficiente.

6. CONTRIBUCIÓN A LA FORMACIÓN PROFESIONAL Y FORMACIÓN GENERAL

Esta disciplina contribuye al logro de los siguientes resultados de la carrera:

- a) Aplicar conocimientos de computación y de matemáticas apropiadas para la disciplina. [Nivel Bloom: 3]
- b) Analizar problemas e identificar y definir los requerimientos computacionales apropiados para su solución. [Nivel Bloom: 4]
- c) Diseñar, implementar y evaluar un sistema, proceso, componente o programa computacional para alcanzar las necesidades deseadas. [Nivel Bloom: 4]
- g) Analizar el impacto local y global de la computación sobre los individuos, organizaciones y sociedad. [Nivel Bloom: 3]
- h) Incorporarse a un proceso de aprendizaje profesional continuo. [Nivel Bloom: 3]
- i) Utilizar técnicas y herramientas actuales necesarias para la práctica de la computación. [Nivel Bloom: 3]
- j) Aplicar la base matemática, principios de algoritmos y la teoría de la Ciencia de la Computación en el modelamiento y diseño de sistemas computacionales de tal manera que demuestre comprensión de los puntos de equilibrio involucrados en la opción escogida. [Nivel Bloom: 4]

7. CONTENIDOS

UNIDAD 1: CN/Computación Paralela.(5 horas)

Nivel Bloom: 4

OBJETIVO GENERAL

- Comparar y contrastar para computación paralela reconociendo las fortalezas y debilidades de cada una.
- Comparar y contrastar paradigma de programación paralela reconociendo las fortalezas y debilidades de cada una.
- Identificar las propiedades básicas de ancho de banda, latencia, escalabilidad, granularidad.
- Diseñar, código, hacer test y depuración de computación paralela.

CONTENIDO

- Revision de los tópicos.
- Models of computation.
- Tipos de computation.
- Tareas paralelas.
- Datos paralelos.
- Cluster.
- Paradigmas de programación paralela.
- Hebras (*threading*).
- Paso de mensajes.
- Técnicas dirigidas a eventos.
- Arquitecturas de software paralelas (*MapReduce*).
- Computación en grilla (*grid computing*).
- Comunidades abiertas de computación distribuida (BOINC, SETI, ...).

Lecturas: [Lin and Snyder, 2008]

UNIDAD 2: AR/Multiprocesamiento.(5 horas)	
Nivel Bloom: 3	
OBJETIVO GENERAL	CONTENIDO
<ul style="list-style-type: none"> ▪ Discutir el concepto de procesamiento paralelo y la relación entre paralelismo y desempeño. ▪ Apreciar que los tipos de datos multimedia (ej. audio y datos visuales de 8/16 bits) pueden ser procesados en paralelo en registros de 64 bits para mejorar el desempeño. ▪ Entender como el desempeño puede ser mejorado incorporando múltiples procesadores en un único chip. ▪ Apreciar la necesidad de expresar algoritmos en una forma que permita la ejecución en procesadores paralelos. ▪ Entender como los procesadores gráficos de propósito especial (GPUs) pueden acelerar el desempeño de aplicaciones gráficas. ▪ Entender la organización de estructuras computacionales que puedan ser electronicamente configuradas y reconfiguradas. 	<ul style="list-style-type: none"> ▪ La ley de Amdahl. ▪ Procesamiento en vectores pequeños (operaciones multimedia). ▪ Procesadores Multinúcleos y Multihebras. ▪ La taxonomía de Flynn: Estructuras multiprocesador y arquitecturas. ▪ Sistemas de programación de múltiples procesadores. ▪ GPU y procesadores gráficos de propósito especial. ▪ Introducción a la lógica reconfigurable y procesadores de propósito especial.
Lecturas: [Lin and Snyder, 2008]	

UNIDAD 3: AL/Algoritmos Paralelos.(3 horas)	
Nivel Bloom: 4	
OBJETIVO GENERAL	CONTENIDO
<ul style="list-style-type: none"> ▪ Describir la implementación de listas enlazadas en un PRAM. ▪ Usar operaciones paralelas para implementar cálculos simples eficientes en paralelo. ▪ Explicar el teorema de Brent y su relevancia. 	<ul style="list-style-type: none"> ▪ El modelo PRAM. ▪ Lecturas y escrituras exclusivas vs concurrentes. ▪ Salto de punteros. ▪ El teorema de Brent y el trabajo eficiente.
Lecturas: [Lin and Snyder, 2008]	

UNIDAD 4: Modelos de Threads con PTHREADs (0 horas)	
Nivel Bloom: 3	
OBJETIVO GENERAL	CONTENIDO
<ul style="list-style-type: none"> ▪ Entender los distintos modelos de programación paralela. ▪ Conocer ventajas y desventajas de los distintos modelos de programación paralela. 	<ul style="list-style-type: none"> ▪ ¿Qué es una hebra? ▪ ¿Qué es pthread? ▪ Diseñando programas con pthreads. ▪ Creación y manejo de hebras. ▪ Sincronización de hebras con mutex.
Lecturas: [Nichols et al., 1996], [Lin and Snyder, 2008]	

UNIDAD 5: Modelos de Threads con OpenMP (0 horas)	
Nivel Bloom: 3	
OBJETIVO GENERAL	CONTENIDO
<ul style="list-style-type: none"> ▪ Implementar programas multihebras por medio de OpenMP. ▪ Entender y aplicar conceptos de sincronización y trabajo compartido. 	<ul style="list-style-type: none"> ▪ ¿Qué es OpenMP? ▪ El modelo de programación OpenMP. ▪ Directivas de OpenMP. ▪ Constructores de trabajo compartido. ▪ Constructores de Tareas. ▪ Constructores de sincronización. ▪ Manejo de datos privados y compartidos.
Lecturas: [Chandra et al., 2000], [Lin and Snyder, 2008]	

UNIDAD 6: Modelo de programación mediante paso de Mensajes con MPI (0 horas)	
Nivel Bloom: 3	
OBJETIVO GENERAL	CONTENIDO
<ul style="list-style-type: none"> ▪ Implementar programas multihebras por medio de OpenMP. ▪ Entender y aplicar conceptos de sincronización y trabajo compartido. 	<ul style="list-style-type: none"> ▪ ¿Qué es MPI? ▪ Rutinas de administración de ambiente. ▪ Rutinas de comunicación punto a punto. ▪ Rutinas de comunicación colectiva. ▪ Tipos de datos derivados. ▪ Rutinas de administración del comunicador y de grupo. ▪ Topología virtual.
Lecturas: [Karniadakis and II, 2003], [Lin and Snyder, 2008]	

UNIDAD 7: <i>Threading Building Blocks (TBB)</i> (0 horas)	
Nivel Bloom: 3	
OBJETIVO GENERAL	CONTENIDO
<ul style="list-style-type: none"> ▪ Entender y aplicar el modelo de datos paralelos utilizando la herramienta TBB. 	<ul style="list-style-type: none"> ▪ Bucles Simples Paralelos. ▪ Bucles Complejos Paralelos. ▪ Cancelación y Exepciones. ▪ Contenedores paralelos.
Lecturas: [Reinders, 2007], [Lin and Snyder, 2008]	

8. METODOLOGÍA

El profesor del curso presentará clases teóricas de los temas señalados en el programa propiciando la intervención de los alumnos.

El profesor del curso presentará demostraciones para fundamentar clases teóricas.

El profesor y los alumnos realizarán prácticas

Los alumnos deberán asistir a clase habiendo leído lo que el profesor va a presentar. De esta manera se facilitará la comprensión y los estudiantes estarán en mejores condiciones de hacer consultas en clase.

9. EVALUACIONES

Evaluación Permanente 1 : 20 %

Examen Parcial : 30 %

Evaluación Permanente 2 : 20 %

Examen Final : 30 %

Referencias

- [Chandra et al., 2000] Chandra, R., Menon, R., Leo Dagum, D. K., Maydan, D., and McDonald, J. (2000). *Parallel Programming in OpenMP*. Morgan Kaufmann, 1 edition.
- [Karniadakis and II, 2003] Karniadakis, G. E. and II, R. M. K. (2003). *Parallel Scientific Computing in C++ and MPI: A Seamless Approach to Parallel Algorithms and their Implementation*. Cambridge University Press, 1 edition.
- [Lin and Snyder, 2008] Lin, C. and Snyder, L. (2008). *Principles of Parallel Programming*. Addison Wesley, 1 edition.
- [Nichols et al., 1996] Nichols, B., Buttlar, D., and Farrell, J. P. (1996). *Pthreads Programming: A POSIX Standard for Better Multiprocessing*. O'Reilly Media, Inc., 1 edition.
- [Reinders, 2007] Reinders, J. (2007). *Intel Threading Building Blocks: Outfitting C++ for Multi-core Processor Parallelism*. O'Reilly Media, Inc., 1 edition.