



Programación Paralela y Concurrente

1. Características generales

Nombre:	Programación Paralela y Concurrente
Sigla:	CI-0117
Créditos:	4
Horas:	5
Requisitos:	CI-0113 Programación 2, MA-0292 Álgebra Lineal para Computación, CI-0114 Fundamentos de Arquitectura
Correquisitos:	CI-0116 Análisis de Algoritmos y Estructuras de Datos
Clasificación:	Curso propio
Ciclo carrera:	II ciclo, 2do año
Docente:	Gr02: Jeisson Hidalgo Céspedes < jeisson.hidalgo@ucr.ac.cr > Gr04: Francisco Arroyo Mora < rafael.arroyo@ucr.ac.cr >
Asistente:	Gr02: Bryan Ulate Caballero < bulatec98@gmail.com > Gr04: Diego Contreras Estrada < diegocontreras2898@gmail.com >
Grupo y semestre:	02, 04, II ciclo 2019
Horario de clase:	Gr02: K 9 a 11:50 102-IF / V 10 a 11:50 302-IF Gr04: K 11 A 12:50 202-IF / V 10 A 12:50 102-IF
Horario de consulta:	Gr02: K 12 a 13 102-IF / V 9 a 10 321-IF2 Gr04: L 13 a 17 240-IF

2. Descripción

La programación concurrente y distribuida de datos y tareas es una ampliación de las habilidades de programación serial desarrolladas en cursos previos. Esta ampliación es imprescindible por cuanto las plataformas de hardware actuales y futuras ofrecen características de computación paralela que no podrían ser aprovechadas por programadores que solo dominen la programación serial.

3. Objetivos

Objetivo General

El objetivo general del curso es resolver problemas que requieren incremento del desempeño mediante la paralelización de datos y tareas a través de los paradigmas de programación concurrente y distribuido.

Objetivos Específicos

Durante el curso el estudiante desarrollará habilidades para:





1. Comprender y explicar las motivaciones y tendencias de la computación paralela para contextualizar el desarrollo de software paralelo en la actualidad mediante el estudio de las características generales de las tecnologías más relevantes.
2. Diseñar soluciones concurrentes y distribuidas correctas.
3. Resolver problemas por paralelización de datos y paralelización de tareas.
4. Construir programas concurrentes para resolver problemas mediante hilos de ejecución y recursos compartidos.
5. Construir programas distribuidos para resolver problemas mediante procesos y recursos distribuidos.
6. Probar programas concurrentes y distribuidos para asegurar su efectividad y calidad mediante el uso de pruebas de software.
7. Evaluar y comparar el desempeño de programas concurrentes y distribuidos para determinar su incremento en el rendimiento respecto a versiones funcionalmente equivalentes mediante la aplicación de métricas básicas de uso común.
8. Explicar el modelo de ejecución concurrente y distribuido (máquina nociónal) para implementar programas de forma correcta.

4. Contenidos

Objetivo	Eje temático	Desglose
1 Comprender motivaciones y tendencias	1.1 La necesidad de computación paralela	Las necesidades de separación de asuntos y desempeño que motiva el software paralelo.
	1.2 Hardware paralelo	Sinopsis de modelos de hardware paralelo. Jerarquía de Flynn
	1.3 Software paralelo	Concepto de proceso y de hilo de ejecución.
2 Diseñar soluciones concurrentes y distribuidas	2.1 Algoritmos paralelos	Diferencia con algoritmos seriales. Análisis espaciotemporal de algoritmos paralelos.
	2.2 Técnicas de descomposición	Descomposición recursiva, de datos, exploratoria, especulativa y otras.
	2.3 Mapeo de tareas a procesos	Características de tareas e interacciones. Técnicas de mapeo para balanceo de carga. Técnicas para reducir la sobrecarga debida a la interacción de tareas.
	2.4 Modelos de programas paralelos	Paralelismo de datos, grafo de tareas, <i>work-pool</i> y otros.





3. Resolver problemas	3.1 Problemas de paralelización de datos y tareas	Resolver problemas como: consumidor-productor, filósofos comensales, problema de los fumadores, problema de la barbería, problema de Santa Claus, formar agua, y el problema de <i>Modus Hall</i> .
4, 8 Construir programas concurrentes con recursos compartidos	4.1 Concurrencia por hilos	Concepto de hilo de ejecución. Espacio de direcciones. Interfaces de programación por hilos (como Pthreads y OpenMP). Rastreo de memoria y procesamiento de hilos de ejecución.
	4.2 Integridad de hilos	Condiciones de carrera (regiones críticas). Código reentrante. Código <i>thread-safe</i> .
	4.3 Mecanismos de sincronización	Espera activa. Mecanismos de sincronización provistos por el API (como <i>mutex</i> , semáforos, candados de lectura-escritura, variables de condición, barreras, reducciones).
5, 8 Construir programas concurrentes con recursos distribuidos	5.1 Concurrencia por procesos	Concepto de proceso. Memoria distribuida. Interfaces de programación por procesos (como MPI). Rastreo de memoria y procesamiento de procesos.
	5.2 Entrada y salida	Entrada y salida mediante procesos paralelos.
	5.3 Comunicación	Comunicación punto a punto y comunicación colectiva entre procesos.
6 Probar	6.1 Pruebas de software	Pruebas de correctitud en programas concurrentes y distribuidos (como caja negra y caja blanca).
7 Evaluar y comparar	7.1 Métricas	Ley de Amdahl. Métricas de aceleración (<i>speedup</i>), eficiencia, y escalabilidad.
	7.2 Desempeño	Medición del tiempo de pared. Gráficos de desempeño.

5. Metodología

Se seguirá una metodología híbrida constructivista y tradicional. Durante las horas extraclase los alumnos estudiarán el material del curso y resolverán los ejercicios planteados. Las soluciones serán presentadas por los estudiantes en un repositorio de control de versiones. Las lecciones serán presenciales en un aula o laboratorio de computadoras de la Escuela. Durante las lecciones se realizarán clases magistrales cortas y actividades de resolución de problemas con el fin de ayudar a los estudiantes a comprender las nociones y resolver los ejercicios planteados y proyectos. Los estudiantes aplicarán las nociones en al menos un proyecto a desarrollar en parejas. Los estudiantes realizarán dos exámenes parciales a responder en papel, en los que podrán consultar material de





apoyo en una hoja carta escrita a mano por ambos lados por el propio estudiante. El profesor podría proponer actividades opcionales por crédito extra en la nota del curso en cualquiera de las evaluaciones.

6. Evaluación

%	Rubro	Descripción
20%	Ejercicios	Ejercicios planteados en el material del curso, agrupados en las cuatro secciones temáticas del cronograma. El tiempo límite de entrega de cada sección será dos semanas después de que se haya cubierto en clases.
25%	Examen 1	Examen estrictamente individual en papel sobre los temas 1 y 2 del cronograma, a realizar dos semanas después de cubiertos los temas.
25%	Examen 2	Examen estrictamente individual en papel de los temas 1 a 4 del cronograma, a realizar dos semanas después de cubiertos los temas.
30%	Proyectos	Problemas para resolver en parejas a presentar en control de versiones, con entregables máximo tres semanas después de cubiertos los temas: 1. Concurrencia compartida imperativa (Pthreads) 2. Concurrencia compartida declarativa (OpenMP) 3. Concurrencia distribuida homogénea (MPI)

7. Cronograma

#	Temática	Semanas	Fechas*
1	Introducción a la concurrencia y distribución Ejes temáticos: 1.1 a 1.3	1	12-ago a 17-ago
2	Concurrencia compartida imperativa (Pthreads) Ejes temáticos: 2.1 a 2.4, 3.1, 4.1 a 4.3, 6.1, 7.1, 7.2	6	19-ago a 28-set
3	Concurrencia compartida declarativa (OpenMP, OpenACC) Ejes temáticos: 2.1 a 2.3, 4.1 a 4.3, 6.1, 7.1, 7.2	4	30-set a 26-oct
4	Concurrencia distribuida (MPI, Qt) Ejes temáticos: 2.2, 2.3, 5.1 a 5.3, 6.1, 7.1, 7.2	5	28-oct a 30-nov

* Fechas tentativas. Se usarán como fechas efectivas de examen, ejercicios y proyectos, las fechas en que se terminen de cubrir los temas respectivos.

8. Bibliografía

1. Pacheco, Peter S. "An introduction to parallel programming". Morgan Kaufmann Pub, 2011.
2. Rauber, Thomas y Rüniger, Gudula. "Parallel Programming (for multicore and cluster systems)". Springer-Verlag Berlin Heigelberg, 2010.
3. Grama, Ananth et.al. "Introduction to Parallel Computing". Addison-Wesley, 2003.





4. Quinn, Michael J. "Parallel Programming in C with MPI and OpenMP". McGraw-Hill Education, 2003.
5. Chandra, Rohit et.al. "Parallel Programming in OpenMP". Morgan Kaufmann Pub, 2001.
6. Downey, Allen B. Little Book of Semaphores.
Disponibile en <https://greenteapress.com/wp/semaphores>

