

Arquitecturas Avanzadas

Código: 102778
Créditos ECTS: 6

Titulación	Tipo	Curso	Semestre
2502441 Ingeniería Informática	OB	3	1
2502441 Ingeniería Informática	OT	4	1

Contacto

Nombre: Juan Carlos Moure Lopez
Correo electrónico: JuanCarlos.Moure@uab.cat

Uso de idiomas

Lengua vehicular mayoritaria: español (spa)
Algún grupo íntegramente en inglés: No
Algún grupo íntegramente en catalán: No
Algún grupo íntegramente en español: No

Equipo docente

Miquel Àngel Senar Rosell

Prerequisitos

Es muy, muy recomendable haber cursado y aprobado las asignaturas de Fundamentos de Informática, Fundamentos de Computadores, Estructura de Computadores, Metodología de la Programación y Arquitectura de Computadores

Objetivos y contextualización

El objetivo fundamental de la asignatura es que los alumnos adquieran las capacidades de análisis, utilización y evaluación de arquitecturas avanzadas de computadores, para el desarrollo de aplicaciones y servicios informáticos.

Los conceptos básicos que se describirán en las sesiones teóricas son: el procesador multi-núcleo, el sistema multi-procesador de memoria compartida, los aceleradores de cómputo, la evaluación de rendimiento y prestaciones, y los modelos y lenguajes de programación de estos sistemas paralelos. Los objetivos más específicos son los siguientes:

1. Entender las claves para obtener buen rendimiento de computadores multi-núcleo y multi-procesador, y de aceleradores de cómputo (GPUs)
2. Identificar las oportunidades de paralelización en un algoritmo o aplicación, a diferentes niveles (instrucciones, iteraciones de un bucle, llamadas a funciones...), y expresarlo formalmente
3. Diseñar las estructuras de datos de una aplicación y los algoritmos que acceden a estos datos para favorecer el buen rendimiento de la jerarquía de memoria
4. Analizar, diseñar e implantar algoritmos paralelos con patrones abstractos de cómputo (reducción, transformación...) bajo paradigmas de programación basados en variables compartidas y sincronización; y con lenguajes paralelos actuales, como OpenMP y OpenACC.

5. Verificar la funcionalidad y evaluar el rendimiento de aplicaciones paralelas, identificando los cuellos de botella respecto al rendimiento

6. A partir del análisis de rendimiento, seleccionar el computador adecuado a una aplicación y/o realizar optimizaciones del código que mejoren aún más su rendimiento

Competencias

Ingeniería Informática

- Actitud personal.
- Adquirir hábitos de trabajo personal.
- Capacidad de analizar y evaluar arquitecturas de computadores, incluyendo plataformas paralelas y distribuidas, así como desarrollar y optimizar software de para las mismas.
- Capacidad para analizar, evaluar, seleccionar y configurar plataformas hardware para el desarrollo y ejecución de aplicaciones y servicios informáticos.
- Capacidad para concebir y desarrollar sistemas o arquitecturas informáticas centralizadas o distribuidas integrando hardware, software y redes.
- Comunicación.

Resultados de aprendizaje

1. Analizar las características de las aplicaciones que requieran altas prestaciones.
2. Analizar los modelos, paradigmas y lenguajes de programación paralela disponibles para determinar el que mejor se adecua a las necesidades de la aplicación.
3. Comunicar eficientemente, de forma oral y/o escrita, conocimientos, resultados y habilidades, tanto en entornos profesionales como ante públicos no expertos.
4. Configurar plataformas hardware y entornos de programación para el desarrollo y ejecución de aplicaciones de altas prestaciones.
5. Desarrollar aplicaciones paralelas basadas en los paradigmas existentes.
6. Determinar las plataformas más adecuadas para cada tipo de aplicación.
7. Evaluar la funcionalidad y el rendimiento de las aplicaciones paralelas/distribuidos desarrolladas.
8. Evaluar las prestaciones de las arquitecturas de altas prestaciones, así como la funcionalidad de las aplicaciones.
9. Evaluar y predecir el rendimiento de distintas plataformas de cómputo de altas prestaciones para la ejecución de aplicaciones.
10. Mantener una actitud proactiva y dinámica respecto al desarrollo de la propia carrera profesional, el crecimiento personal y la formación continuada. Espíritu de superación.
11. Prevenir y solucionar problemas.
12. Proponer soluciones informáticas basadas en sistemas de cómputo de altas prestaciones que integren tanto la arquitectura de los componentes hardware del sistema, así como lo interconexión de los mismos, y el diseño del software necesario.

Contenido

1. Paralelismo de Instrucciones

- Reordenación dinámica de la ejecución de las instrucciones
- Dependencias de datos entre instrucciones en un bucle

2. Paralelismo de Datos

- Análisis del paralelismo de datos de un algoritmo
- Paralelismo SIMD (vectorización) y MIMD (multi-*thread*)
- Arquitecturas Vectoriales e instrucciones SIMD
- GPUs y aceleradores de cómputo

3. Arquitectura de computadores paralelos

- Estructura y arquitectura de los procesadores multi-hilo y multi-núcleo, y de los nodos multi-procesador
- Jerarquía de memoria (NUMA) y soporte para la coherencia de datos en *cache*
- Soporte H/W para la sincronización y para la comunicación entre *threads*

4. Algoritmos Paralelos

- Modelo de paralelismo con variables compartidas y sincronización. Patrones de cómputo paralelo
- Paralelismo de datos (SIMD): operaciones de reducción y de transformación sobre estructuras de datos multi-dimensionales
- Paralelismo de tareas (MIMD): *master/worker*, *divide & conquer*, *pipeline*
- Programación no determinista y sincronización sin *locks*
- Directivas OpenMP y OpenACC

5. Evaluación del Rendimiento de Aplicaciones Paralelas

- Análisis de complejidad, paralelismo y localidad
- Medida del tiempo de ejecución, IPC, ancho de banda e intensidad aritmética
- Trabajo total y camino crítico de la ejecución paralela
- *Speedup* y eficiencia de la ejecución paralela
- Sobrecarga (*overhead*) de la sincronización y de las comunicaciones

Metodología

Clases de Teoría: se expondrán los conocimientos propios de la asignatura. Se describirán los conceptos básicos y se ilustrarán con ejemplos prácticos. Se resaltarán los problemas de aprendizaje más importantes y se mostrará cómo completar y profundizar en los contenidos. Se discutirán casos prácticos y el profesor detectará los problemas de comprensión y razonamiento más comunes y los resolverá para todos los estudiantes.

Clases de Problemas en Laboratorio: se harán actividades cooperativas de resolución de problemas y de casos prácticos sencillos, que servirán de apoyo a la teoría. Tras un trabajo individual previo, los alumnos harán una puesta en común en grupo y resolverán sus dudas. El profesor detectará los problemas de comprensión y razonamiento más comunes y los resolverá por grupos o para todos los estudiantes. Se promueve la capacidad de análisis y síntesis del alumno, el razonamiento crítico y la capacidad de resolución de problemas. Durante la sesión los alumnos deberán informar al profesor sobre sus avances y los problemas que se puedan encontrar, y entregarán un documento final con los resultados y reflexiones sobre sus actividades.

Caso práctico de Ingeniería de Rendimiento: durante todo el curso, los alumnos deberán diseñar, planificar, realizar, presentar y defender de forma oral un trabajo práctico, indicando los objetivos, el desarrollo, los resultados obtenidos con gráficas comprensibles, las incidencias más notables y, finalmente, las conclusiones más relevantes. Se espera por parte de los alumnos una actitud proactiva y dinámica, la búsqueda autónoma de los conocimientos que sean necesarios, y ambición con los objetivos del trabajo.

Competencias Transversales:

- T02.04 - Prevenir y solucionar problemas. Se trabaja en las clases de problemas en el laboratorio y se evalúa con los problemas que los alumnos han de resolver en grupo.
- T04.01 - Comunicar eficientemente, oralmente o por escrito, conocimientos, resultados y habilidades, tanto en entornos profesionales como frente a públicos no expertos. Se trabaja en las tutorías y se evalúa en las entregas de trabajos y en la exposición del trabajo práctico que han de hacer los alumnos, y supone un 15% de la nota final.
- T06.01 - Mantener una actitud proactiva y dinámica respecto al desarrollo de la propia carrera profesional, el crecimiento personal y la formación continuada. Tener espíritu de superación. Se incentiva y se evalúa en las sesiones de prácticas y en la preparación del trabajo de Ingeniería de Rendimiento, donde el profesor interactúa con los alumnos y observa y ayuda a corregir sus actitudes; supone un 25% de la nota de estas actividades.

Actividades

Título	Horas	ECTS	Resultados de aprendizaje
Tipo: Dirigidas			
Problemas en el Laboratorio	24	0,96	2, 1, 9, 7, 8, 4, 5, 6, 10, 11
Teoría	24	0,96	2, 1, 8, 4, 6, 12
Tutorías Aula	6	0,24	2, 8, 4, 12
Tipo: Supervisadas			
Preparación del trabajo a realizar en el laboratorio	15	0,6	9, 7, 5, 10
Preparación y Exposición del Trabajo Práctico	5	0,2	3
Tipo: Autónomas			
Estudio autónomo	24	0,96	2, 8, 4, 6, 12
Preparación de sesiones prácticas	30	1,2	2, 1, 5
Resolución de Problemas	20	0,8	1, 7, 8, 6, 11

Evaluación

Las fechas de evaluación continua y entrega de trabajos se publicarán en la plataforma Moodle de la UAB (Campus Virtual) y pueden estar sujetas a cambios de programación por motivos de adaptación a posibles incidencias. Siempre se informará en la plataforma Moodle sobre estos cambios ya que se entiende que es la plataforma habitual de intercambio de información entre profesor y estudiantes.

Actividades de evaluación

Actividad A: resolución de problemas prácticos en el laboratorio; se realiza en grupo; supone un 25% de la nota final; es necesario obtener una nota mínima de 3 puntos sobre 10; y no se puede recuperar.

Actividad B: entrega de un informe de la actividad A; se realiza individualmente; supone un 5% de la nota final; no es necesario obtener una nota mínima; y no se puede recuperar.

Actividad C: propuesta y resolución de un caso práctico de Ingeniería de Rendimiento; se realiza en grupo; supone un 35% de la nota final; es necesario obtener una nota mínima de 3 puntos sobre 10; y se puede recuperar.

Actividad D: defensa oral de la actividad C; se realiza individualmente; supone un 5% de la nota final; no es necesario obtener una nota mínima; y se puede recuperar.

Actividad E: entrega de un informe de la actividad C; se realiza en grupo; supone un 5% de la nota final; no es necesario obtener una nota mínima; y no se puede recuperar.

Actividad F: asistencia y participación activa en clase; se realiza individualmente; supone un 5% de la nota final; no es necesario obtener una nota mínima; y no se puede recuperar.

Actividad G: examen escrito de teoría y problemas; se realiza individualmente y sin apuntes; supone un 20% de la nota final; es necesario obtener una nota mínima de 3 puntos sobre 10; y se puede recuperar.

Programa de Actividades de Evaluación

Las actividades A, C y F se realizan durante todo el curso. Las actividades B, D y E se realizan al final del curso. La actividad G se realiza en las fechas especificadas por la coordinación.

Proceso de Recuperación

El estudiante se puede presentar a la recuperación siempre que se haya presentado a las actividades B, D y G. De acuerdo con la coordinación del Grado y la dirección de la Escuela de Ingeniería solamente se podrán recuperar las actividades C, D y G, que representan el 60% de la nota final.

Procedimiento de Revisión

Todas las actividades de evaluación se podrán revisar en un lugar, fecha y hora que se publicará con anterioridad. Se podrán hacer reclamaciones sobre la nota de la actividad, que serán evaluadas por el profesorado responsable de la asignatura. Si el estudiante no se presenta a esta revisión, no se revisará posteriormente esta actividad.

Calificaciones

Se otorgará una calificación de matrícula de honor (MH) a los estudiantes que alcancen una nota final igual o superior a 9,00, hasta un total del 5% de los estudiantes matriculados, priorizando la parte de la nota que corresponde a evaluaciones individuales (actividades B, D, F y G).

Si el alumno o alumna no alcanza la nota mínima en alguna de las actividades de evaluación, la calificación numérica final será el mínimo entre 4,5 puntos y la suma ponderada de las notas de todas las actividades.

Un estudiante se considerará no evaluable (NA) si no se ha presentado a ninguna de las actividades B, D y G.

Irregularidades por parte de los estudiantes, copia y plagio

Sin perjuicio de otras medidas disciplinarias que se estimen oportunas, se calificarán con un cero la irregularidades cometidas por un estudiante que puedan conducir a una variación de la calificación de un acto de evaluación. Por tanto, la copia, el plagio, el engaño, dejar copiar, etc. en cualquiera de las actividades de evaluación implicará suspenderla con un cero. Las actividades de evaluación calificadas de esta forma y por este procedimiento no serán recuperables. Si es necesario superar cualquiera de estas actividades de evaluación para aprobar la asignatura, entonces esta asignatura quedará suspendida directamente, sin oportunidad de recuperarla en el mismo curso. En el caso de no superar la asignatura debido a que el estudiante ha cometido irregularidades en un acto de evaluación, la nota numérica del expediente será el valor menor entre 3.0 y la media ponderada de las notas (y por tanto no será posible el aprobado por compensación).

Evaluación de los estudiantes repetidores

No hay ningún tratamiento especial para los estudiantes repetidores.

Actividades de evaluación

Título	Peso	Horas	ECTS	Resultados de aprendizaje
Asistencia y participación activa en clase	5%	0	0	3, 10
Defensa oral del trabajo de Ingeniería de Rendimiento	5%	0,1	0	3
Entrega de un informe de la resolución de problemas prácticos en el laboratorio	5%	0	0	3
Entrega de un informe del caso práctico de Ingeniería de Rendimiento	5%	0	0	3
Examen escrito individual de teoría y problemas	20%	1,9	0,08	2, 1, 7, 3, 6, 11, 12
Propuesta y resolución de un caso práctico de Ingeniería de Rendimiento	35%	0	0	2, 1, 9, 7, 8, 4, 5, 6, 10, 11, 12
Resolución de problemas prácticos en el laboratorio	25%	0	0	1, 9, 7, 8, 4, 5, 6, 10, 11, 12

Bibliografía

Computer Architecture: A Quantitative Approach. 6th Edition
John Hennessy, David Patterson, Morgan Kaufmann (Elsevier) 2018 (Cap. 4 y 5)

Structured Parallel Programming: Patterns for efficient computation
M. McCool, J. Reinders, A. Robison, Elsevier, 2012

Parallel Programming for Multicore and Cluster Systems
T. Rauber, G. Runger, Springer (Elsevier), 2010

Programming Massively Parallel Processors: A Hands-on Approach
D. Kirk, & W.M. Hwu, Morgan Kaufmann (Elsevier), 2010