

Guía docente de la asignatura

Asignatura	COMPUTACIÓN PARALELA Y MODELOS EMERGENTES	
Materia	SISTEMAS Y SERVICIOS EMPOTRADOS, UBICUOS Y DE ALTAS PRESTACIONES	
Módulo	(vacío)	
Titulación	MASTER EN INGENIERÍA INFORMÁTICA	
Plan	510	Código 53170
Periodo de impartición	2º CUATRIMESTRE	Tipo/Carácter Obligatoria
Nivel/Ciclo	MASTER	Curso 2014-2015
Créditos ECTS	6 ECTS	
Lengua en que se imparte	CASTELLANO	
Profesor/es responsable/s	AGUSTÍN DE DIOS HERNÁNDEZ (responsable teoría y prácticas) ARTURO GONZÁLEZ ESCRIBANO	
Datos de contacto (E-mail, teléfono...)	E-MAIL: agustin@infor.uva.es , E-MAIL: arturo@infor.uva.es ,	
Horario de tutorías	Copiar literalmente lo siguiente: Véase www.uva.es → Centros → Campus de Valladolid → Escuela Técnica Superior de Ingeniería Informática → Tutorías	
Departamento	DEPARTAMENTO DE INFORMÁTICA	

1. Situación / Sentido de la Asignatura

1.1 Contextualización

Dentro de la perspectiva de los estudios de un máster sobre informática de perfil profesionalizante, en esta asignatura se pretende aportar conocimientos sobre conceptos de programación para la obtención de altas prestaciones, modelos de programación en plataformas de computo paralelo, modelos de programación de GP-GPUs, el soporte para programación a nivel de sistema de los procesadores actuales y los modelos de programación SIMD en procesadores de altas prestaciones.

1.2 Relación con otras materias

La materia guarda relación con asignaturas de programación, estudio de sistemas operativos y arquitectura de computadores.

1.3 Prerrequisitos

Conocimientos sobre programación, sistemas operativos y arquitectura de computadores a nivel de Grado en Informática.

2. Competencias

2.1 Generales

Código	Descripción
CG1	Capacidad para proyectar, calcular y diseñar productos, procesos e instalaciones en todos los ámbitos de la ingeniería informática
CG2	Capacidad para la dirección de obras e instalaciones de sistemas informáticos, cumpliendo la normativa vigente y asegurando la calidad del servicio.
CG4	Capacidad para el modelado matemático, cálculo y simulación en centros tecnológicos y de ingeniería de empresa, particularmente en tareas de investigación, desarrollo e innovación en todos los ámbitos relacionados con la Ingeniería en Informática.
CG6	Capacidad para la dirección general, dirección técnica y dirección de proyectos de investigación, desarrollo e innovación, en empresas y centros tecnológicos, en el ámbito de la Ingeniería Informática.
CG7	Capacidad para la puesta en marcha, dirección y gestión de procesos de fabricación de equipos informáticos, con garantía de la seguridad para las personas y bienes, la calidad final de los productos y su homologación.
CG8	Capacidad para la aplicación de los conocimientos adquiridos y de resolver problemas en entornos nuevos o poco conocidos dentro de contextos más amplios y multidisciplinares, siendo capaces de integrar estos conocimientos.

2.2 Específicas

Código	Descripción
CET1	Capacidad para modelar, diseñar, definir la arquitectura, implantar, gestionar, operar, administrar y mantener aplicaciones, redes, sistemas, servicios y contenidos informáticos.
CET6	Capacidad para diseñar y evaluar sistemas operativos y servidores, y aplicaciones y sistemas basados en computación distribuida.
CET7	Capacidad para comprender y poder aplicar conocimientos avanzados de computación de altas prestaciones y métodos numéricos o computacionales a problemas de ingeniería.
CET8	Capacidad de diseñar y desarrollar sistemas, aplicaciones y servicios informáticos en sistemas empotrados y ubicuos.
CET11	Capacidad para conceptualizar, diseñar, desarrollar y evaluar la interacción persona-ordenador de productos, sistemas, aplicaciones y servicios informáticos.

3. Objetivos

Comprender los principios de la programación de altas prestaciones, en plataformas de computo no convencionales o con nuevos modelos de programación que explotan las capacidades de paralelismo de los sistemas actuales y futuros. Asimilar los problemas y retos de la programación paralela y de alto rendimiento en

diferentes niveles de abstracción y detalle. Desarrollar habilidades y dominar técnicas de programación para el desarrollo de aplicaciones y servicios con necesidades de alto rendimiento en arquitecturas reales.

4. Tabla de dedicación del estudiante a la asignatura

ACTIVIDADES PRESENCIALES	HORAS	ACTIVIDADES NO PRESENCIALES	HORAS
Clases teórico-prácticas (T/M)	22	Estudio y trabajo autónomo individual	90
Clases prácticas de aula (A)	8	Estudio y trabajo autónomo grupal	0
Laboratorios (L)	30		
Prácticas externas, clínicas o de campo	0		
Seminarios (S)	0		
Tutorías grupales (TG)	0		
Evaluación (fuera del periodo oficial de exámenes)	0		
Total presencial	60	Total no presencial	90

5. Bloques temáticos

Bloque 1: Modelos de computación paralela

Carga de trabajo en créditos ECTS: |

a. Contextualización y justificación

El desarrollo de aplicaciones eficientes, en cualquier campo de la ciencia y la ingeniería, que exploten el potencial de cómputo de las nuevas plataformas y arquitecturas actuales, implica la utilización de técnicas y modelos de programación que exploten paralelismo. En este primer bloque se expondrá una visión general de los diferentes modelos y aproximaciones de programación sobre nuevas plataformas paralelas que permitan al alumno aproximarse a los nuevos modelos y técnicas de programación desde una perspectiva multinivel.

b. Objetivos de aprendizaje

Comprender los problemas inherentes a la programación paralela en diferentes niveles de abstracción. Entender los problemas conceptuales y las soluciones aportadas por diferentes modelos de programación.

c. Contenidos

TEMA 1: Modelos de computación paralela

- 1.1 Modelos de sistemas paralelos
- 1.2 Algoritmia paralela y estudio de rendimiento
- 1.3 Modelos de programación paralela

d. Métodos docentes

Actividades presenciales. En cada sesión se realizarán diferentes actividades dirigidas a facilitar la adquisición de conocimientos, habilidades y competencias:

- Clase magistral participativa
- Tareas de búsqueda, procesado y evaluación de información
- Resolución de problemas de forma individual y por grupos
- Discusión crítica y evaluación de resultados

Actividades no presenciales: El alumno realizará una serie de actividades fuera del aula:

- Preparación de sesiones presenciales: Lectura de bibliografía, búsquedas de información, preparación de dudas.
- Repaso de conceptos y ejercicios de consolidación
- Ejercicios de aprendizaje cooperativo

e. Plan de trabajo

Se organizará en unas 10 horas en sesiones presenciales a realizar en el laboratorio. En cada sesión se realizarán diferentes actividades.

f. Evaluación

Véase el punto 7 sobre evaluación

g. Bibliografía básica

Introducción a la Programación Paralela. Francisco Almeida et.al. Paraninfo 2008, ISBN 978-84-9732-674-2

h. Bibliografía complementaria

Introduction to Parallel Computing (2nd Edition), Grama et.al. 2003, ISBN 0-201-64865-2

i. Recursos necesarios

Laboratorio de ordenadores del centro. Cada ordenador tiene el software y recursos necesarios para la ejecución de los ejercicios y consultas a través de la red.

Plataforma de docencia virtual Moodle: Campus virtual de la UVa (<http://campusvirtual.uva.es/>)

a. Contextualización y justificación

Los sistemas GPU ofrecen un elevado potencial de rendimiento especialmente dentro del campo de las aplicaciones de cálculo científicas y de todos los campos de la ingeniería. El trabajo con estas unidades presenta unas características distintivas tanto a nivel de su programación como el de la gestión de la memoria sobre la que trabajan.

b. Objetivos de aprendizaje

Asimilar las técnicas básicas de programación de propósito general en dispositivos GPU. Entender las interrelaciones entre el sistema de memoria, la programación y el rendimiento obtenido en estos dispositivos. Utilizar técnicas básicas de optimización de código para dispositivos GPU.

c. Contenidos

TEMA 2: Programación de propósito general en dispositivos GPU

- 1.1 Dispositivos GPU: Modelo abstracto
- 1.2 Modelos y programación: CUDA, OpenCL, OpenACC
- 1.3 Técnicas básicas de programación GP-GPU
- 1.4 Problemas de rendimiento y técnicas básicas de optimización

d. Métodos docentes

Actividades presenciales. En cada sesión se realizarán diferentes actividades dirigidas a facilitar la adquisición de conocimientos, habilidades y competencias:

- Clase magistral participativa
- Resolución de problemas de forma individual y por grupos
- Ejercicios de programación

Actividades no presenciales: El alumno realizará una serie de actividades fuera del aula:

- Preparación de sesiones presenciales: Lectura de bibliografía, búsquedas de información, preparación de dudas.
- Repaso de conceptos y ejercicios de consolidación

e. Plan de trabajo

Se organizará en unas 10 horas en sesiones presenciales a realizar en el laboratorio. En cada sesión se realizarán diferentes actividades.

f. Evaluación

Véase el punto 7 sobre evaluación

g. Bibliografía básica

- Programming Massively Parallel Processors: A Hands-on Approach, David B. Kirk y Wen-mei W. Hwu. Morgan Kaufmann, 2010. ISBN: 978-0-12-381472-2, 1
- CUDA by Example: An Introduction to General-Purpose GPU Programming, Jason Sanders y Edward Kandrot. Addison-Wesley Professional, 2010. ISBN: 0131387685

h. Bibliografía complementaria

i. Recursos necesarios

Laboratorio de ordenadores del centro. Cada ordenador tiene el software y recursos necesarios para la ejecución de los ejercicios y consultas a través de la red.

Plataforma de docencia virtual Moodle: Campus virtual de la UVa (<http://campusvirtual.uva.es/>)

Bloque 3: Soporte de programación de sistema en arquitecturas de 32 y 64 bits

Carga de trabajo en créditos ECTS: |

a. Contextualización y justificación

Los modernos procesadores de 32 y 64 bits ofrecen recurso de programación a nivel de sistema que van más allá de las meras extensiones de los ya conocidos. En particular son de resaltar todos los recursos para la gestión avanzada de excepciones y los recursos para entornos multiprocesador, de continua utilización en la administración de los procesadores multicore. Y también la gestión de otros aspectos, tales como la administración del consumo de potencia y la regulación térmica de los núcleos.

b. Objetivos de aprendizaje

- Conocer la arquitectura de sistema IA-32 e IA-32e que soportan un entorno de programación de 64 bits. Ser capaz de crear entornos fiables y seguros para programas de aplicación.
- Conocer la gestión de interrupciones y excepciones de la arquitectura de 64 bits de forma que sea capaz de generar programas de aplicación y de sistema con un control exacto de los diferentes errores y situaciones de sistema que puedan surgir. Conocer la organización y el funcionamiento del APIC.
- Conocer los elementos de la arquitectura que dan soporte a múltiples procesadores, entre ellos, las operaciones atómicas, la consistencia de memoria, la inicialización de múltiples procesadores y la arquitectura multi-core. Conocer y poder manejar de forma práctica aquéllas consideraciones de programación relevantes para el hardware.
- Conocer los procedimientos hardware y software de gestión e inicialización del procesador.
- Conocer los mecanismos hardware y el interfaz software del procesador para la gestión de la temperatura, la energía y su relación con las prestaciones.

c. Contenidos

TEMA 3. Soporte de programación de sistema en arquitecturas de 64 y 32 bits

3.1 Revisión general de la arquitectura de sistema

3.2 La gestión de interrupciones y excepciones. APIC.

3.3 Soporte para gestión de multiprocesadores

3.4 Soporte para gestión e inicialización del procesador

3.5 Soporte para gestión de potencia y control de temperatura

d. Métodos docentes

- Clase magistral participativa
- Estudio de casos en aula y en laboratorio
- Resolución de problemas

e. Plan de trabajo

Se organizará en unas 7 horas de clase de teoría, 3 horas de clase de problemas y 10 horas de prácticas en laboratorio.

f. Evaluación

Véase el punto 7 sobre evaluación

g. Bibliografía básica

- Intel 64 and IA-32 Architectures Software Developer Manual. Intel. 2013.
 - Capítulos 2, 6, 8, 9 y 14

h. Bibliografía complementaria

No se aporta

i. Recursos necesarios

Libro de texto, presentaciones audiovisuales, recursos cooperativos de visualización para el aprendizaje, resolución de problemas.

Bloque 4: Soporte para procesamiento SIMD en los procesadores actuales: las extensiones AVX de Intel

Carga de trabajo en créditos ECTS: |

a. Contextualización y justificación

La necesidad de un mayor rendimiento computacional en todos los segmentos de la industria ha llevado a la aparición de nuevas extensiones a los repertorios conocidos aportando procesamiento vectorial con instrucciones de 256 bits. Estas instrucciones permiten mejorar la administración de datos y el rendimiento de aplicaciones para fines generales, como el procesamiento de imágenes, sonido y vídeo, simulaciones científicas, análisis financieros y modelos y análisis tridimensionales en todos los campos de la ingeniería.

b. Objetivos de aprendizaje

- Conocer el modelo de programación basado en las extensiones SIMD de los procesadores actuales
- Conocer las instrucciones más relevantes de las extensiones SIMD que se utilizan para realizar cálculos en ciencias e ingeniería. En concreto las de las instrucciones AVX de Intel.

c. Contenidos

TEMA 4. Soporte para procesamiento SIMD en los procesadores actuales: las extensiones AVX de Intel

- 4.1 Introducción
- 4.2 El modelo de programación a nivel de aplicación y de sistema
- 4.3 Instrucciones básicas: aritméticas, lógicas, comparación
- 4.4 Instrucciones FMA
- 4.5 Instrucciones de sincronización transaccional
- 4.6 Otras instrucciones

d. Métodos docentes

- Clase magistral participativa
- Estudio de casos en aula y en laboratorio
- Resolución de problemas

e. Plan de trabajo

Se organizará en unas 7 horas de clase de teoría, 3 horas de clase de problemas y 10 horas de prácticas en laboratorio.

f. Evaluación

Véase el punto 7 sobre evaluación

g. Bibliografía básica

- Intel Architecture: Instrucción Set Extensions Programming Reference. Intel. 2012

h. Bibliografía complementaria

No se aporta

i. Recursos necesarios

Libro de texto, presentaciones audiovisuales, recursos cooperativos de visualización para el aprendizaje, resolución de problemas.

6. Temporalización (por bloques temáticos)

El número de semanas de un cuatrimestre son 15. Ejemplo:

BLOQUE TEMÁTICO	CARGA ECTS	PERIODO PREVISTO DE DESARROLLO
Bloque 1: Modelos de computación paralela	1,0 ECTS	Semanas 1 a 3
Bloque 2: Modelos de programación GP-GPU	1,0 ECTS	Semanas 4 a 5
Bloque 3: Soporte de programación de sistema en arquitecturas de 32 y 64 bits	2,0 ECTS	Semanas 6 a 10
Bloque 4: Soporte para procesamiento SIMD en los procesadores actuales: las extensiones AVX de Intel	2,0 ECTS	Semanas 11 a 15

7. Sistema de calificaciones – Tabla resumen

INSTRUMENTO/PROCEDIMIENTO	PESO EN LA NOTA FINAL	OBSERVACIONES
Examen final escrito	100 %	Periodo de exámenes

Tanto en la convocatoria ordinaria como en la extraordinaria se realizará un examen escrito, el cual constará de dos partes:

- Primera parte: problemas cortos y cuestiones sobre los contenidos teóricos, a desarrollar en 90 minutos. Valoración: 75 %. En concreto 7,5 puntos como máximo de un total de 10.
- Segunda parte: cuestiones cortas sobre los tratados en las prácticas de laboratorio, a desarrollar en 60 minutos. Valoración: 25 %. En concreto 2,5 puntos como máximo de un total de 10.

CRITERIOS DE CALIFICACIÓN
<ul style="list-style-type: none">• Convocatoria ordinaria y extraordinaria: A la hora de calificar las pruebas se considerarán fundamentalmente los siguientes aspectos<ul style="list-style-type: none">○ Correcta utilización de conceptos, definiciones y propiedades relacionadas con la naturaleza de la situación que se trata de resolver o explicar.○ Justificaciones teóricas que se aporten para el desarrollo de las respuestas. Se penalizará la no justificación, ausencia de explicaciones o explicaciones incorrectas.○ Claridad y coherencia en la exposición.○ Precisión en cálculos y notaciones.○ Cada ejercicio se valorará de acuerdo a lo estipulado en el enunciado del examen.

8. Anexo: Métodos docentes

Actividad	Metodología
Clase de teoría	En estas sesiones, de forma motivadora y que mueva al alumno a su implicación personal, se tratará de dirigir a éste hacia los conceptos claves y se le iniciará en el planteamiento de los principales problemas. En todo momento se tratará de utilizar un enfoque de conocimiento de concepto orientado a la resolución de problemas prácticos
Clase práctica	En las prácticas de laboratorio supervisadas se pretende que el alumno realice un primer contacto directo con los conceptos abordados en la materia tal como éstos aparecen en un entorno profesional. En las sesiones prácticas se le plantearán al alumno casos concretos que debe resolver haciendo uso de herramientas profesionales. Fundamentalmente estarán orientadas a la familiarización con herramientas orientadas a posibilitar la medida del rendimiento de la ejecución de las aplicaciones informáticas bajo sus diversos aspectos.
Tutoría	En las sesiones de tutoría cada alumno podrá plantear personalmente al profesor de la asignatura cuantas cuestiones le hayan surgido durante su trabajo con la materia. Asimismo, el profesor dará a cada estudiante los consejos que considere oportunos para ayudarle a obtener un aprovechamiento óptimo en el proceso de adquisición de competencias que corresponde a esta materia.

9. Anexo: Cronograma de actividades previstas

Se ha aportado ya en el plan de trabajo