

PLAN DOCENTE DE LA ASIGNATURA

Curso académico: 2026/2027

Identificación y características de la asignatura					
Código	402422				
Denominación (español)	Computación Paralela				
Denominación (inglés)	Parallel Computing				
Titulaciones	Máster Universitario en Ingeniería Informática				
Centro	Escuela Politécnica				
Módulo	Computación de altas prestaciones				
Materia	Computación de altas prestaciones				
Carácter	Obligatoria	ECTS	3	Semestre	1
Profesorado					
Nombre		Despacho		Correo-e	
Juan Mario Haut Hurtado		EI 17		juanmariohaut@unex.es	
Área de conocimiento	Arquitectura y Tecnología de Computadores				
Departamento	Tecnología de los Computadores y de las Comunicaciones				
Profesor/a coordinador/a (si hay más de uno)	Juan Mario Haut Hurtado				
Competencias / Resultados de aprendizaje					
Habilidades o destrezas: HB05 - Al finalizar el aprendizaje, el o la estudiante será capaz de analizar un problema de ingeniería para aplicar los conceptos avanzados de computación de altas prestaciones y métodos numéricos necesarios para resolverlo. TIPO: Habilidades o destrezas.					
Contenidos					
Descripción general del contenido: Diseñar algoritmos paralelos usando paradigmas de computación de memoria compartida y memoria distribuida en entornos multi-núcleo/multi-procesador, multi-nodo y heterogéneos.					

Temario

Denominación del tema 1: Fundamentos y diseño de algoritmos paralelos.

Contenidos del tema 1: Motivación, fuentes y niveles de paralelismo. Modelos de memoria compartida, distribuida y entornos heterogéneos. Descomposición de datos y tareas; dependencias, granularidad y balanceo de carga. Aceleración, eficiencia y límites de paralelización; interpretación de la ley de Amdahl y de la escalabilidad fuerte y débil. Protocolo básico de medida y validación numérica. El análisis del rendimiento se retomará en los temas siguientes.

Descripción de las actividades prácticas del tema 1: Familiarización con el entorno de trabajo y ejecución de código de referencia proporcionado. Identificación de regiones paralelizables, comprobación de resultados y medida de tiempos mediante un guion. Interpretación de datos de escalabilidad de una implementación ya disponible, sin exigir programar todavía con OpenMP, MPI o CUDA.

Denominación del tema 2: Programación paralela en memoria compartida.

Contenidos del tema 2: Modelo de ejecución en sistemas multinúcleo/multiprocesador. OpenMP o modelo equivalente: regiones paralelas, bucles y reparto de trabajo. Variables compartidas y privadas. Condiciones de carrera, sincronización, exclusión mutua y reducciones. Planificación de iteraciones y balanceo de carga. Introducción al paralelismo de tareas. Optimización básica centrada en la corrección, los sobrecostes de sincronización y el acceso a memoria.

Descripción de las actividades prácticas del tema 2: Paralelización progresiva de un caso numérico a partir de código secuencial y ejemplos guiados. Detección y corrección de una condición de carrera; aplicación de una reducción y comparación de alternativas de reparto de trabajo. Validación, medida de aceleración y explicación breve de los cuellos de botella observados.

Denominación del tema 3: Programación paralela en memoria distribuida.

Contenidos del tema 3: Modelo de paso de mensajes: procesos, rangos y comunicadores básicos. Comunicaciones punto a punto y prevención de interbloqueos. Operaciones colectivas de difusión, reparto, recogida y reducción. Distribución de datos y balanceo entre procesos. Costes de comunicación y análisis de escalabilidad en entornos multinodo. Introducción guiada a las comunicaciones no bloqueantes, su finalización y las posibilidades de solapamiento con el cómputo.

Descripción de las actividades prácticas del tema 3: Adaptación del caso numérico a MPI o modelo equivalente, con código inicial y guion de trabajo. Distribución de datos, cálculo local y composición del resultado. Validación y experimentos de escalabilidad fuerte. Ejemplo guiado de comunicación no bloqueante e interpretación de resultados de escalabilidad débil, sin exigir un segundo proyecto.

Denominación del tema 4: Programación paralela en arquitecturas heterogéneas.

Contenidos del tema 4: Aceleradores y GPU para cálculo general. Introducción al modelo de ejecución: funciones de cómputo (kernels), hilos y bloques. Jerarquía de memoria a nivel básico. Coordinación CPU/GPU, transferencias y selección de las partes del algoritmo que conviene acelerar. Evaluación del tiempo de cálculo y del tiempo total, incluidas las transferencias. El alcance se limita a una GPU y una única tecnología de programación.

Descripción de las actividades prácticas del tema 4: Adaptación de un núcleo numérico sencillo a partir de una plantilla, con CUDA, OpenCL o tecnología equivalente seleccionada para el curso. Comprobación del resultado frente a la versión CPU; medida separada del cálculo y del tiempo total. Comparación al variar el tamaño del problema y la configuración de ejecución, justificando las ventajas o limitaciones observadas.

Actividades formativas

TEMA	TOTAL	Actividades Presenciales (AP)					Actividades Virtuales (AV)				TP	TA
		GG	CH	L	O	S	CST	CSP	CAT	CAP		
1	10	0	0	0	0	0	1	0	2	1	0	6
2	21	0	0	0	0	0	2	2	2	2	0	13
3	24	0	0	0	0	0	2	3	2	2	0	15
4	18	0	0	0	0	0	1	2	2	2	0	11
Evaluación	2	2	0	0	0	0	0	0	0	0	0	0
Totales	75	2	0	0	0	0	6	7	8	7	0	45
% Presencialidad							% Virtualidad					
6,7 %							93,3 %					

Actividades Presenciales (AP)

Actividades que se desarrollan en un único espacio físico y que implican interacción física entre estudiante y docente:

- GG: Grupo Grande (85 estudiantes).
- CH: Actividades de prácticas clínicas hospitalarias (7 estudiantes)
- L: Actividades de laboratorio o prácticas de campo (15 estudiantes)
- O: Actividades en sala de ordenadores o laboratorio de idiomas (20 estudiantes)
- S: Actividades de seminario o de problemas en clase (40 estudiantes).

Actividades Virtuales (AV)

Actividades que no se desarrollan en un espacio físico común. Pueden ser síncronas (implican interacción estudiante / docente) o asíncronas:

- CST: Clase síncrona teórica.
- CSP: Clase síncrona práctica.
- CAT: Clase asíncrona teórica.
- CAP: Clase asíncrona práctica.

- TP: Tutorías Programadas (seguimiento docente, tutorías ECTS).
- TA: Trabajo autónomo del estudiante.

Metodologías docentes

1. Docencia basada en clases expositivas para la presentación de los contenidos de la asignatura.
2. Actividades prácticas, sesiones de laboratorio guiadas, seminarios de resolución de problemas, etc. en grupos bajo la dirección de un profesor.
3. Seguimiento síncrono del aprendizaje del estudiante.
4. Realización de actividades, trabajos y estudio por parte del estudiante.

5. Metodologías centradas en el alumnado como el Aprendizaje basado en Proyectos (ABP), Aprendizaje basado en Equipos (TBL), Aula Invertida, Aprendizaje Cooperativo, Gamificación, etc.

Aplicación en la asignatura: Las prácticas se organizarán en una actividad inicial guiada (tema 1) y tres entregas incrementales (temas 2 a 4), reutilizando el caso numérico, el código secuencial y los criterios de validación y medida. No se añadirá un proyecto final independiente.

Las actividades asíncronas guiadas dispondrán de instrucciones, ejemplos y retroalimentación. El trabajo autónomo incluye estudio, desarrollo independiente, informes breves y preparación del examen, sin contabilizar de nuevo las horas de las actividades guiadas.

Sistemas de evaluación
1. Examen presencial (35 %): prueba objetiva, semiojetiva, de desarrollo escrito/oral y/o resolución de problemas sobre los conocimientos adquiridos. Se realizará presencialmente en las instalaciones del centro. 2. Evaluación de trabajos dirigidos y evidencias (65 %): valoración de prácticas de programación paralela, informes técnicos, entregas y otras evidencias vinculadas al trabajo desarrollado durante la asignatura.
Resultados de aprendizaje
HB05 - Al finalizar el aprendizaje, el o la estudiante será capaz de analizar un problema de ingeniería para aplicar los conceptos avanzados de computación de altas prestaciones y métodos numéricos necesarios para resolverlo. TIPO: Habilidades o destrezas.
Bibliografía (básica y complementaria)
Básica: P. Pacheco y M. Malensek, An Introduction to Parallel Programming, 2.^a ed., Morgan Kaufmann. A. Grama, A. Gupta, G. Karypis y V. Kumar, Introduction to Parallel Computing, 2.^a ed., Addison-Wesley. Complementaria y normativa técnica: OpenMP Architecture Review Board, OpenMP Application Programming Interface Specification. MPI Forum, MPI: A Message-Passing Interface Standard. B. R. Gaster et al., Heterogeneous Computing with OpenCL, Morgan Kaufmann. NVIDIA, CUDA C++ Best Practices Guide (documentación oficial).
Otros recursos y materiales docentes complementarios
Campus Virtual de la UEx para materiales, actividades asíncronas, entregas, retroalimentación y comunicación. Guiones secuenciados, código inicial en C/C++, casos de prueba, instrucciones de compilación y ejecución, y herramientas de medida del rendimiento. Se presupone manejo básico de programación, arrays y funciones; no dominio previo de las tecnologías paralelas utilizadas. Recursos necesarios para las prácticas: sistema multinúcleo, entorno MPI con acceso multinodo y acceso a una GPU compatible con la tecnología seleccionada. Su disponibilidad y forma de acceso deberán concretarse antes de las prácticas; no se presupone que el estudiante tenga GPU propia.