



Guía docente				
Datos Identificativos				2018/19
Asignatura (*)	Programación Paralela Avanzada	Código	614973107	
Titulación	Mestrado Universitario en Computación de Altas Prestacións / High Performance Computing (Mod. Virtual 2018)			
Descriptorios				
Ciclo	Periodo	Curso	Tipo	Créditos
Máster Oficial	2º cuatrimestre	Primero	Optativa	6
Idioma	CastellanoGallegoInglés			
Modalidad docente	Presencial			
Prerrequisitos				
Departamento	Enxeñaría de Computadores			
Coordinador/a	Fraguela Rodriguez, Basilio Bernardo	Correo electrónico	basilio.fraguela@udc.es	
Profesorado	Darriba López, Diego	Correo electrónico	diego.darriba@udc.es	
	Fraguela Rodriguez, Basilio Bernardo		basilio.fraguela@udc.es	
Web	aula.cesga.es			
Descripción general	<p>En esta materia se incrementarán los conocimientos de programación paralela adquiridos por los alumnos en el cuatrimestre anterior en las asignaturas "Programación paralela" y "Programación de arquitecturas heterogéneas". El objetivo será que los alumnos aprendan a optimizar códigos paralelos para grandes arquitecturas paralelas o supercomputadores actuales, usando como base para sus pruebas los recursos proporcionados por el Centro de Supercomputación de Galicia (CESGA) y el Grupo de Arquitectura de Computadores (GAC) de la Universidade da Coruña (UDC).</p> <p>Se centrará en aquellos aspectos de las aplicaciones paralelas que suelen penalizar el rendimiento, como son las comunicaciones, el balanceo de carga, el acceso a memoria o el manejo de entrada/salida. También se abordará la computación multiplataforma que permita aprovechar el paralelismo a nivel de tareas entre varios aceleradores hardware, así como la computación híbrida donde una misma aplicación haga uso de varios paradigmas de programación paralela de cara a obtener un buen rendimiento en clústers de sistemas multinúcleo y/o aceleradores hardware.</p>			

Competencias del título	
Código	Competencias del título
A1	CE1 - Definir, evaluar y seleccionar la arquitectura y el software más adecuado para la resolución de un problema
A2	CE2 - Analizar y mejorar el rendimiento de una arquitectura o un software dado
A4	CE4 - Profundizar en el conocimiento de herramientas de programación y diferentes lenguajes en el campo de la computación de altas prestaciones
A5	CE5 - Analizar, diseñar e implementar algoritmos y aplicaciones paralelas eficientes
A7	CE7 - Conocer las arquitecturas emergentes en el campo de la supercomputación
B1	CB6 - Poseer y comprender conocimientos que aporten una base u oportunidad de ser originales en el desarrollo y/o aplicación de ideas, a menudo en un contexto de investigación
B2	CB7 - Que los estudiantes sepan aplicar los conocimientos adquiridos y su capacidad de resolución de problemas en entornos nuevos o poco conocidos dentro de contextos más amplios (o multidisciplinares) relacionados con su área de estudio
B5	CB10 - Que los estudiantes posean las habilidades de aprendizaje que les permitan continuar estudiando de un modo que habrá de ser en gran medida autodirigido o autónomo.
B6	CG1 - Ser capaz de buscar y seleccionar la información útil necesaria para resolver problemas complejos, manejando con soltura las fuentes bibliográficas del campo
B9	CG4 - Ser capaz de planificar y realizar tareas de investigación, desarrollo e innovación en ámbitos relacionados con la computación de altas prestaciones
C1	CT1 - Utilizar las herramientas básicas de las tecnologías de la información y las comunicaciones (TIC) necesarias para el ejercicio de su profesión y para el aprendizaje a lo largo de su vida

Resultados de aprendizaje
---------------------------



Resultados de aprendizaje	Competencias del título		
Conocer optimizaciones avanzadas aplicables a programas paralelos	AP1 AP2 AP5	BP1 BP2 BP5 BP6 BP9	
Controlar la afinidad y el balanceo de carga	AP5	BP1 BP2 BP5 BP6 BP9	CP1
Optimizar comunicaciones en sistemas de memoria distribuida	AP2 AP4 AP5	BP1 BP2 BP5 BP6 BP9	CP1
Realizar entrada/salida paralela	AP4 AP5	BP1 BP2 BP5 BP6 BP9	CP1
Programar sistemas con varios aceleradores hardware	AP4 AP5 AP7	BP1 BP2 BP5 BP6 BP9	CP1
Programar sistemas con memoria compartida/distribuida	AP4 AP5 AP7	BP1 BP2 BP5 BP6 BP9	CP1

Contenidos	
Tema	Subtema
1- Técnicas avanzadas de optimización de códigos paralelos.	-
2- Control de afinidad y balanceo de carga.	-
3- Optimización de comunicaciones en sistemas de memoria distribuida.	-
4- Entrada/salida paralela.	-
5- Programación híbrida para sistemas con varios aceleradores hardware.	-
6- Programación híbrida para sistemas de memoria compartida/distribuida.	-

Planificación				
Metodologías / pruebas	Competencias	Horas presenciales	Horas no presenciales / trabajo autónomo	Horas totales



Prácticas de laboratorio	A2 A5 C1	4	80	84
Trabajos tutelados	A1 A2 A4 A5 A7 B1 B2 B5 B6 B9 C1	0	45	45
Prueba mixta	A5 A2 B2	2	0	2
Lecturas	A1 A4 A7 B1	0	18	18
Atención personalizada		1	0	1

(\*Los datos que aparecen en la tabla de planificación són de carácter orientativo, considerando la heterogeneidad de los alumnos

Metodologías	
Metodologías	Descripción
Prácticas de laboratorio	En esta actividad se realizan tareas autónomas, aunque con instrucciones del profesorado, que permiten al alumno familiarizarse desde un punto de vista práctico con los contenidos expuestos en los materiales de lectura sobre los contenidos de la materia.
Trabajos tutelados	Realización de trabajos en los que el alumno tiene que emplear los conocimientos adquiridos para resolver distintos problemas de forma autónoma.
Prueba mixta	Realización de una prueba de evaluación de la materia.
Lecturas	Lectura y visionado de material relativo al contenido de cada tema. El alumno dispondrá de todo el material necesario según el calendario de la asignatura. El profesor promoverá una actitud activa, promoviendo que el estudiante formule preguntas que permitan aclarar aspectos concretos. El material dejará cuestiones abiertas para la reflexión del alumno.

Atención personalizada	
Metodologías	Descripción
Trabajos tutelados Prácticas de laboratorio	Tanto en las prácticas realizadas autónomamente así como durante el desarrollo de los trabajos tutelados, los estudiantes podrán presentar cuestiones, dudas, etc. El profesor/a, atendiendo a estas solicitudes, repasará conceptos, resolverá nuevos problemas o utilizará cualquier actividad que considere adecuada para resolver las cuestiones planteadas.

Evaluación			
Metodologías	Competencias	Descripción	Calificación
Trabajos tutelados	A1 A2 A4 A5 A7 B1 B2 B5 B6 B9 C1	Calidad del trabajo y del progreso del alumno durante su realización	70
Prueba mixta	A5 A2 B2	Corrección y calidad de las soluciones propuestas por los estudiantes a las cuestiones planteadas en la prueba	30

Observaciones evaluación
<p>En las actividades de evaluación a distancia se les podrá requerir a los alumnos la aplicación de mecanismos que garanticen su identidad así como la autoría de los elementos evaluables presentados.</p> <p>Todas las actividades de evaluación recogidas en esta guía conforman el proceso de evaluación continua de la asignatura. Ni las clases ni la mayor parte de las actividades de evaluación requieren presencialidad del alumno, siendo la excepción la prueba mixta, de un máximo de 2 horas. Esto, unido al hecho de que todos los materiales de la asignatura están disponibles en la plataforma web de educación del título, favorece el trabajo y la evaluación de los alumnos matriculados a tiempo parcial y con dispensa académica de exención de docencia.</p>

Fuentes de información



<b>Básica</b>	- Using Advanced MPI: Modern Features of the Message-Passing Interface. 2014. W. Gropp, T. Hoefler, R. Thakur, E. Lusk. MIT Press- Using OpenMP: The Next Step: Affinity, Accelerators, Tasking, and SIMD (Scientific and Engineering Computation). 2017. R. van der Pas, E. Stotzer, C. Terboven. MIT Press- OpenCL Programming Guide. 2011. A. Munshi, B. Gaster, T. G. Mattson, J. Fung, D. Ginsburg. Addison-Wesley/Pearson Education- Using Advanced MPI: Modern Features of the Message-Passing Interface. 2014. W. Gropp, T. Hoefler, R. Thakur, E. Lusk. MIT Press- Using OpenMP: The Next Step: Affinity, Accelerators, Tasking, and SIMD (Scientific and Engineering Computation). 2017. R. van der Pas, E. Stotzer, C. Terboven. MIT Press- OpenCL Programming Guide. 2011. A. Munshi, B. Gaster, T. G. Mattson, J. Fung, D. Ginsburg. Addison-Wesley/Pearson Education
<b>Complementaria</b>	- Multi-core programming. 2006. S. Akhter e J. Roberts. Intel Press. - Professional CUDA C Programming. 2014. J. Cheng, M. Grossman, T. McKercher. Wross.- Multi-core programming. 2006. S. Akhter e J. Roberts. Intel Press. - Professional CUDA C Programming. 2014. J. Cheng, M. Grossman, T. McKercher. Wross.

### Recomendaciones

#### Asignaturas que se recomienda haber cursado previamente

Programación de Arquitecturas Heterogéneas/614473103

Programación Paralela/614473102

#### Asignaturas que se recomienda cursar simultáneamente

#### Asignaturas que continúan el temario

Trabajo Fin de Máster/614473111

### Otros comentarios

Debido a la fuerte interrelación entre la parte teórica y la parte práctica, y a la progresividad en la presentación de conceptos muy relacionados entre sí en la parte teórica, es recomendable dedicar un tiempo de estudio o repaso diario. En esta materia se hará un uso intensivo de herramientas de comunicación on line: videoconferencia, correo electrónico, chat, etc.

(\*) La Guía Docente es el documento donde se visualiza la propuesta académica de la UDC. Este documento es público y no se puede modificar, salvo cosas excepcionales bajo la revisión del órgano competente de acuerdo a la normativa vigente que establece el proceso de elaboración de guías