

Programación Distribuida y Paralela

4º Ingeniería Superior en Informática



***Depto. de Lenguajes y Sistemas Informáticos
Universidad de Granada***



Datos de la Asignatura

PÁGINA WEB: <http://lsi.ugr.es/~jmantas/pdp/pdp.html>

Profesor de teoría: José Miguel Mantas Ruiz (jmmantas@ugr.es)

- <http://lsi.ugr.es/~jmantas/>
- **Despacho:** 20, planta 3ª
 - **Tutorías:** Lunes de 11 a 12, Miércoles de 09 a 10 y de 11:00 a 13:00 y Viernes de 9:00 a 10:00 y de 12:00 a 13:00.

Profesor de prácticas: Fco. Javier Melero Rus (fjmelero@ugr.es)

- **Despacho:** 20, planta 3ª
- **Tutorías:** Jueves de 12 a 14 y Viernes de 10 a 12

Horario de teoría: Martes de 12:00 a 14:00 en Aula 1.2.

Horarios de prácticas: Aula 2.2.

- **Grupo 1:** Viernes de 16:00 a 18:00.
- **Grupo 2:** Viernes de 18:00 a 20:00.

Introducción

- Existen aplicaciones con **elevados requisitos de cómputo** que requieren software para plataformas multiprocesador.
- **Auge de sistemas paralelos:**
 - Procesadores multicore y manycore
 - Redes de ordenadores
- El procesamiento paralelo está al alcance de cualquiera.
- El **aprovechamiento** del potencial de los **sistemas paralelos** requiere disponer de fundamentos de programación paralela y distribuida.
- Nos centraremos en la programación de **clusters de ordenadores**, los **sistemas multinúcleo** y los **procesadores gráficos (GPUs)**.

Objetivos

1. Dotar al alumno de un conocimiento general sobre programación paralela y sistemas paralelos.
2. Conocer distintos modelos de programación y técnicas para el diseño, evaluación e implementación de algoritmos paralelos.
3. Dotar al alumno de cierta experiencia en la resolución paralela y distribuida de problemas con objeto de mejorar las prestaciones.
4. Conocer los entornos de programación más difundidos para el desarrollo de sistemas paralelos.

Programa de Teoría

Tema 1: INTRODUCCIÓN A LA PROG. DISTRIBUIDA Y PARALELA

Tema 2: METODOLOGÍA DE PROGRAMACIÓN PARALELA

Tema 3: NOTACIONES DE PROGRAMACIÓN PARALELA

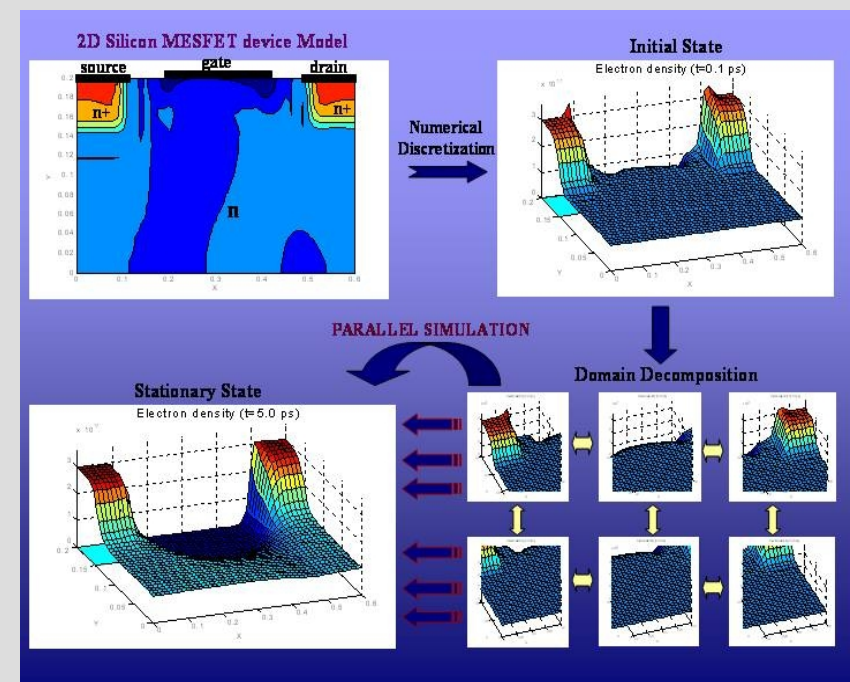
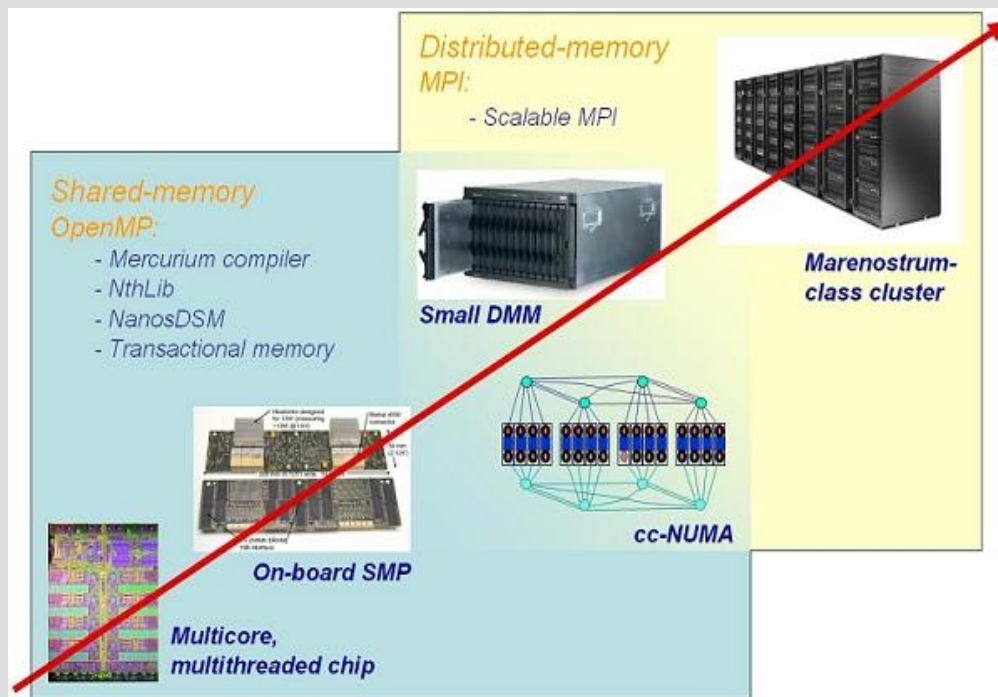
Tema 1: INTRODUCCIÓN A LA PROGRAMACIÓN DISTRIBUIDA Y PARALELA

1.1: Motivación y aspectos de la Programación Paralela

1.2: Modelos de Sistemas Paralelos

1.3: Modelos de Programación Paralela.

1.4: Evaluación del rendimiento de Programas Paralelos

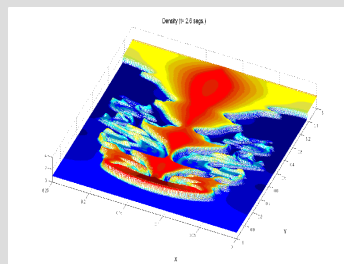


Tema 2: METODOLOGÍA DE PROGRAMACIÓN PARALELA

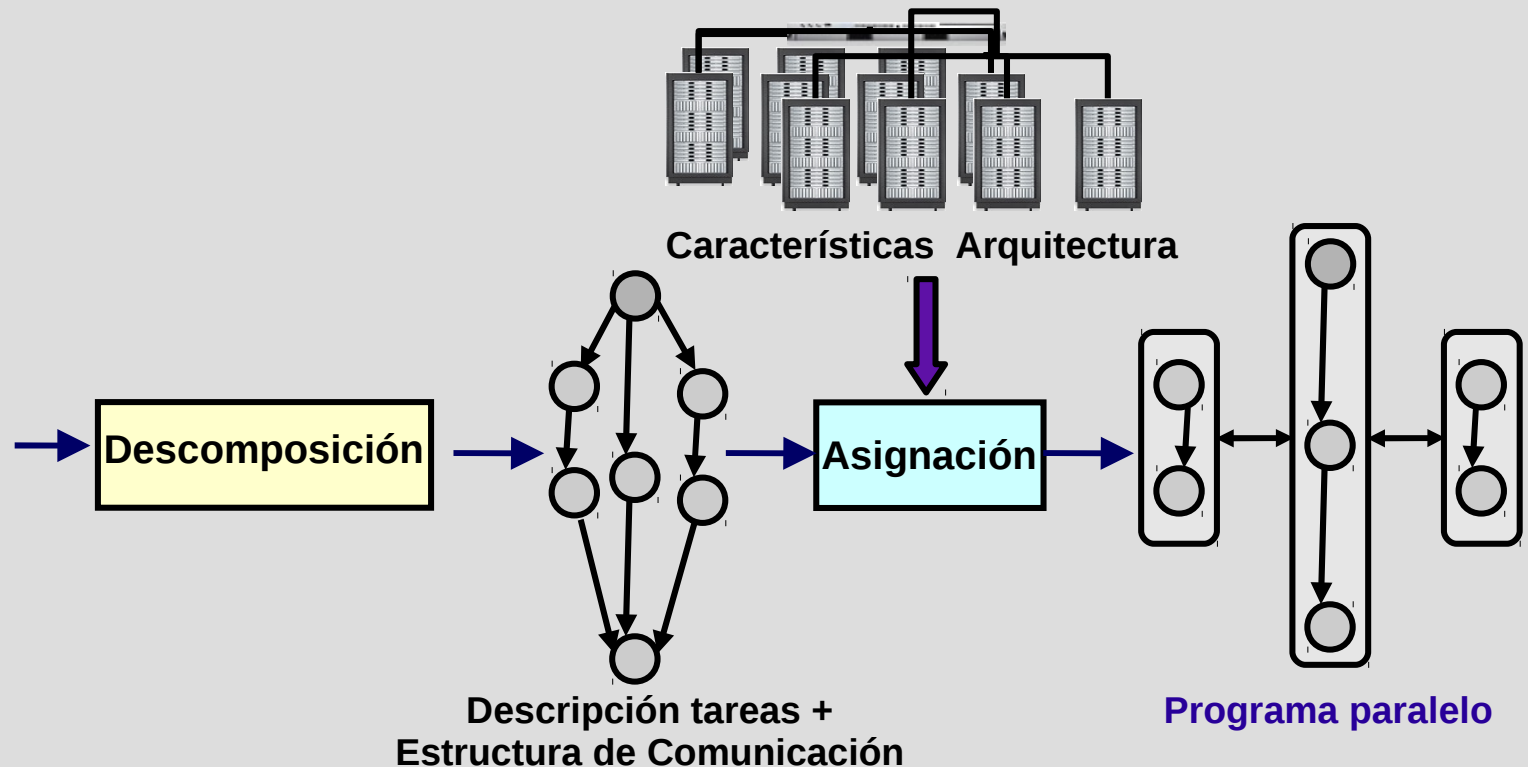
2.1: Descomposición

2.2: Asignación

2.3: Estudio de casos prácticos

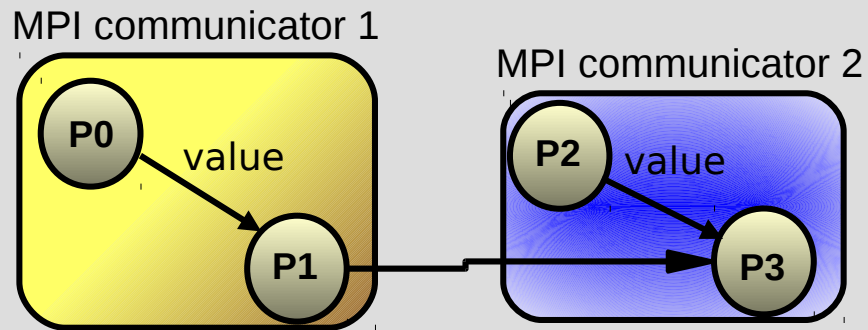


Problema a paralelizar

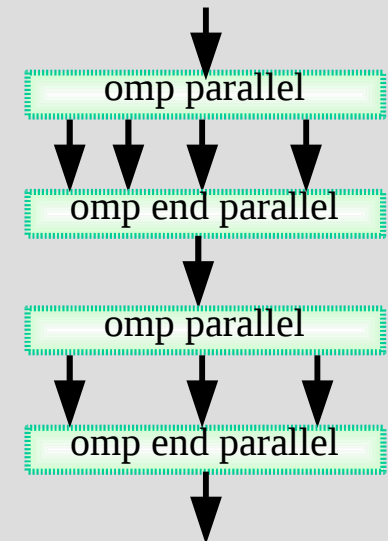


Tema 3: NOTACIONES DE PROGRAMACIÓN PARALELA

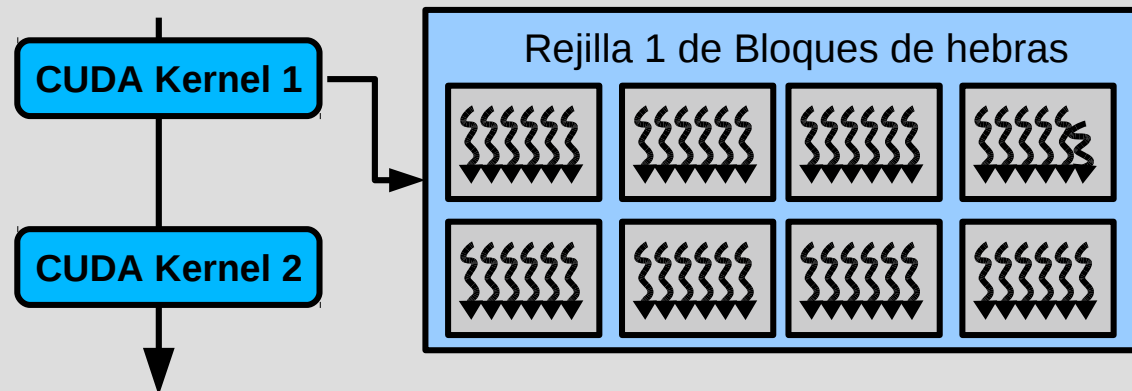
3.1: La Interfaz de Paso de Mensajes: MPI.



3.2: Programación paralela con hebras basada en directivas: OpenMP.



3.3: Introducción a la programación de hardware gráfico paralelo. CUDA

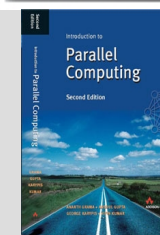
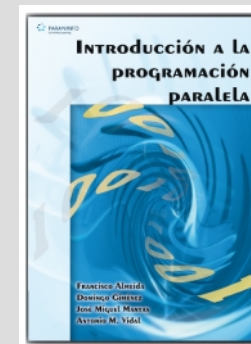


Programa de Prácticas

- **Práctica 1: Introducción a la interfaz de paso de mensajes MPI**
 - Tutorial interactivo de MPI
- **Práctica 2: Implementación paralela de algoritmos de análisis de grafos:**
 - Algoritmo de Floyd Descomposición 1D y 2D
- **Práctica 3: Implementación paralela de un algoritmo de ramificación-acotación:**
 - Resolución distribuida del problema del Viajante de Comercio
- **Práctica 4 : Introducción a la programación en OpenMP**
 - Tutorial interactivo de OpenMP

Bibliografía

- **Introduccion a la Programacion Paralela.** F. Almeida, D. Gimenez, Jose Miguel Mantas, A.M. Vidal. Paraninfo Cengage Learning, 2008.
- **Introduction to Parallel Computing.** Kumar, V., Grama, A., Gupta, A., Karypis G. Benjamin/Cummings Publishing Company, 2003.
- **Parallel Programming. Techniques and applications using networked workstations and parallel computers. Vol II.** Wilkinson, B., Allen, M. Prentice-Hall. 2005.
- **Parallel Programming in C with MPI and OpenMP.** Michael J. Quinn. McGraw-Hill, 2003.
- **The Art of Concurrency: A Thread Monkey's Guide to Writing Parallel Applications.** Clay Breshears. O'Reilly Media, 2009.
- **Página Oficial de Open MPI:** <http://www.open-mpi.org/>
- **Página Oficial de OpenMP:** <http://www.openmp.org>
- **NVIDIA CUDA 3.0 Programming Guide.** NVIDIA 2010



Sistema de Evaluación

- La teoría y las prácticas de la asignatura se evalúan por separado. La teoría y las prácticas tienen el **mismo peso** en la nota final de la asignatura (50 %). Para aprobar la asignatura es necesario tener una **nota media superior o igual a 5** (sobre 10). Para hacer media entre teoría y prácticas es necesario obtener **al menos un 3** (sobre 10) **en cada parte**. La calificación de la parte aprobada se guarda hasta la convocatoria de Diciembre del curso siguiente.
- La teoría se evaluará mediante un examen final. No obstante, existe la posibilidad de superar la teoría mediante **exámenes parciales eliminatorios** que se realizarán para cada tema o realizando un trabajo.
- Se dará la posibilidad de realizar un **trabajo teórico-práctico** sobre algún tópico relacionado con la asignatura que el profesor estime interesante para subir la nota final o superar temas del programa de teoría. Los trabajos se asignan antes de diciembre.
- Se valorará la **participación activa** del alumno en clase a través de la realización de ejercicios y presentación de trabajos sobre los contenidos de la asignatura.

Normas para la realización y evaluación de las prácticas

- Para evaluar la parte de prácticas, el alumno deberá **entregar la documentación** requerida en cada guión de las prácticas obligatorias. Para que dicha documentación pueda ser evaluada, será necesaria la asistencia a algunas de las sesiones de prácticas dentro de los límites de entrega de dicha práctica.
- La superación de la práctica 1 es obligatoria pero no contribuye a la nota de prácticas. Las prácticas 2, 3 y 4 contribuirán con un 35%, 40% y 25% respectivamente en la nota final de prácticas.
- Las prácticas se podrán realizar por grupos de dos personas
- **Primera sesión práctica:** 5 de octubre de 2011.

Fechas límite de entrega de cada práctica:

- Práctica 1: 26 Oct. 2012
- Práctica 2: 30 Nov. 2012
- Práctica 3: 11 Enero 2013
- Práctica 4: 25 Enero 2013

Temas de interés para trabajos

- Grid Computing, Cloud Computing
- Entornos, lenguajes y herramientas para programar procesadores multinúcleo.
- Frameworks y lenguajes para programación paralela con procesadores gráficos (GPUs).
- Intel Threading Building Blocks.
- Los clusters de ordenadores como plataforma de procesamiento paralelo.
- Cuestiones de rendimiento en OpenMP.
- Herramientas de Análisis del rendimiento de programas paralelos (depuradores, trazadores, etc.).
- Programación paralela en sistemas heterogéneos. Open Computing Language.
- Desarrollo de algoritmos paralelos para alguna área de interés (Procesamiento de imágenes, optimización, simulación, ordenación, minería de datos, datawarehouse, etc.).
- Algún tópico relacionado con la asignatura que proponga el alumno.