

**NOTA IMPORTANTE:** Es fundamental leer atentamente los enunciados de los ejercicios, para realizarlos correctamente. Los fallos derivados de no haber leído el enunciado se penalizarán fuertemente en la nota de la práctica.

## 1. Introducción

En esta práctica se propone paralelizar un algoritmo utilizando OpenMP y analizar su rendimiento. Para ello, es necesario aplicar todas las técnicas estudiadas a lo largo del módulo, de cara a justificar las decisiones tomadas en la paralelización, explicar el rendimiento obtenido y mejorar la paralelización propuesta.

## 2. Objetivos

Mediante el desarrollo de esta práctica se pretenden cubrir los siguientes objetivos concretos:

- Analizar un programa para decidir qué regiones son candidatas a ser paralelizadas.
- Paralelizar el programa usando el paradigma que más oportuno se considere.
- Analizar el rendimiento de una implementación paralela, incluyendo su escalabilidad.

## 3. Algoritmos propuestos

Para el desarrollo de la práctica, se proponen los siguientes algoritmos, cada uno con una nota máxima en función de su dificultad. Escoge el algoritmo que prefieras y descarga el código base correspondiente de la plataforma Moodle.

- **MD** (Nota máxima 7): Una implementación sencilla de un algoritmo de dinámica molecular. Partiendo del estado inicial de un conjunto de moléculas, calcula la evolución de este a lo largo del tiempo considerando las interacciones moleculares. Es necesario utilizar el flag `-lm` para compilar. Las variables `np` y `step_num` controlan el tamaño de problema.
- **Poisson** (Nota máxima 7): Resuelve la ecuación de poisson, relacionada con el operador de poisson y muchos problemas físicos. Es necesario utilizar el flag `-lm` para compilar. La variable `tolerance` controla el tamaño del problema.

- **FFT** (Nota máxima 8.5): Implementación sencilla de la transformada de Fourier. Es necesario utilizar el flag `-lm` para compilar. La variable `ln2_max` controla el tamaño del problema.
- **SparseLU** (Nota máxima 8.5): Realiza la factorización LU de una matriz dispersa. Las constantes `DIM` y `BLOCK` controlan el tamaño del problema.
- **Health** (Nota máxima 10): Simula la ocupación de un sistema hospitalario a lo largo del tiempo en función del número y gravedad de pacientes. Las variables globales de la sección **SIMULATION DEFINITION** controlan el tamaño del problema. Muy en particular `sim_level`, `sim_cities` y `sim_time`.
- **Heated Plate** (Nota máxima 10): Calcula la difusión de calor a lo largo del tiempo en una región rectangular. El tamaño de problema lo controlan las constantes `M` y `N` y la variable `epsilon`.

## 4. Realización de la práctica

El objetivo general es maximizar el rendimiento de la paralelización. Para ello, es fundamental el análisis de la aplicación, la realización de pruebas de rendimiento y la toma de decisiones basadas en los resultados. Al igual que en las otras prácticas, recuerda que es importante utilizar tamaños de programa que produzcan un tiempo de ejecución secuencial suficientemente largo como para justificar la ejecución paralela. En la evaluación de esta práctica, más allá del propio código, las justificaciones de las decisiones tomadas y la explicación de los resultados obtenidos tendrán un peso muy importante. La expectativa es que el informe responda preguntas como:

- ¿Por qué he decidido paralelizar esta(s) parte(s) del código?
- ¿Por qué he escogido este paradigma de paralelización?
- ¿Por qué mi paralelización es la óptima? ¿Qué decisiones he tomado para conseguirlo? Si no he conseguido tanto rendimiento como esperaba, ¿por qué?
- ¿Escala adecuadamente mi paralelización? ¿Por qué?

Estas preguntas no son un guión para realizar la entrega, sino ejemplos de contenidos relevantes. El informe podrá incluir figuras, tablas o cualquier material que se considere relevante de cara a justificar las decisiones tomadas y demostrar que el alumno maneja con soltura los conceptos estudiados en el módulo. Si las pruebas de rendimiento no se realizan en los equipos del laboratorio, debe indicarse claramente el procesador en el que se han realizado, así como su número de cores. También debes indicar claramente los tamaños de problema utilizados en los experimentos.

## 5. Evaluación

La evaluación de estas prácticas se realizará mediante una única entrega, que tendrá como fecha límite el día **5 de Diciembre de 2023**. Esta entrega constará de una memoria en formato **pdf** siguiendo la plantilla proporcionada. En esta memoria se recogerán los resultados de todos los ejercicios propuestos en las diferentes prácticas. Es fundamental **explicar los resultados y conclusiones obtenidas**, ya que en caso contrario se penalizará fuertemente la nota. Adicionalmente, se deben entregar los ficheros con el código fuente de los programas desarrollados. La entrega del informe y los programas se realizará en un único fichero comprimido en **zip**, a través de la plataforma Moodle.