

Informe de Prácticas

Sesión 2

Computación de Altas Prestaciones

Máster en Ingeniería Informática



Universidad de Oviedo

Autor:

Rubén Martínez Ginzo, UO282651@uniovi.es

Abril 2025

Índice

1. Introducción	4
1.1. <i>Desarrollo</i>	4
1.2. <i>Benchmarking</i>	4
2. Fase 1	5
3. Fase 2	6
4. Fase 3	7
5. Comparativa	8

Índice de figuras

Índice de tablas

1. Introducción

En esta sesión de prácticas de laboratorio se evalúa la hibridación de lenguajes para la implementación de la multiplicación de matrices, combinando específicamente *C* y *Python*. La experimentación se divide en tres fases:

- **Fase 1:** Implementación de la multiplicación de matrices en *Python* y toma de tiempos.
- **Fase 2:** Creación y manejo de matrices con *Python*, multiplicación de matrices en *C* y toma de tiempos.
- **Fase 3:** Creación y manejo de matrices con *C*, multiplicación de matrices en *C* y toma de tiempos.

Utilizando la librería *ctypes* se lleva a cabo la hibridación de ambos lenguajes, con el objetivo de evaluar el rendimiento de cada una de las fases y comparar los resultados obtenidos en cada uno de los escenarios. La multiplicación de matrices se realizará siguiendo los diferentes esquemas de acceso a memoria evaluados en la sesión anterior: *Row-Major order*, *Column-Major order* y *Z-Order*.

1.1. Desarrollo

Para el desarrollo de esta práctica, se han seguido las indicaciones recogidas en el guion de la sesión correspondiente. Se han implementado las tres fases del experimento, y se han llevado a cabo diversas pruebas para medir su rendimiento. Con el fin de comparar los tiempos de ejecución, se ha desarrollado un programa en *Python* con fines de *benchmarking*, tal y como se detalla en el capítulo siguiente. Todo el código fuente se encuentra disponible públicamente en el siguiente repositorio de GitHub, así como en el archivo *zip* asociado a esta entrega.

1.2. Benchmarking

Como parte complementaria al desarrollo de la sesión de laboratorio, se ha implementado un programa en *Python* que automatiza la medición de los tiempos de ejecución correspondientes a cada una de las fases del experimento.

Este programa, ubicado en el subdirectorio *benchmark*, permite especificar diferentes tamaños de matriz y un número fijo de iteraciones para cada prueba. Una vez configurado, ejecuta secuencialmente cada una de las fases, registrando el tiempo medio de ejecución para la multiplicación de matrices en cada caso. Para facilitar esta tarea, en cada uno de los programas desarrollados en esta práctica (*multiply_matrices.py*, *multiply_matrices_hybrid.py* y *multiply_matrices_hybrid_pro.py*) se han definido las funciones *run_phase_X_row_col* y *run_phase_X_zorder*, donde *X* indica el número de fase. Estas funciones llevan a cabo la toma de tiempos de ejecución de cada producto en la fase correspondiente con los esquemas de acceso a memoria *Row-Major order*, *Column-Major order* y *Z-Order*.

Una vez completadas todas las iteraciones, el programa genera un *log* en formato CSV que recoge los resultados obtenidos. Dichos tiempos de ejecución constituyen la base de los análisis y comparativas que se presentan en los capítulos siguientes.

2. Fase 1

3. Fase 2

4. Fase 3

5. Comparativa