

Taller Evaluación de Rendimiento

Sistemas Operativos

Jose David Ontiveros Gutiérrez
Pontificia Universidad Javeriana

Abril 2025

1. Introducción

El informe presenta un taller centrado en evaluar el rendimiento de un algoritmo clásico de multiplicación de matrices, ejecutado en modo secuencial y paralelo, usando 1, 2 y 4 hilos. El objetivo es analizar cómo la paralelización y diferentes configuraciones de hardware y software afectan el tiempo de ejecución en tareas computacionalmente intensivas. Se espera que el rendimiento mejore al aumentar el número de hilos, especialmente con matrices grandes. Los resultados permitirán concluir si la paralelización reduce efectivamente el tiempo de ejecución.

2. Descripción del Taller

El propósito de la evaluación es analizar el impacto de la paralelización en la eficiencia del algoritmo, examinando cómo esta puede acelerar el tiempo de ejecución en distintas plataformas de hardware. Comparar las ejecuciones secuenciales y paralelas facilitará la identificación de mejoras en el rendimiento y permitirá comprobar la influencia de la configuración del sistema en los resultados obtenidos.

2.1. Compilación y Ejecución

Una vez documentado y preparado el script `lanza.pl`, se procedió a compilar y ejecutar el programa siguiendo los pasos descritos a continuación:

- Compilación de los archivos fuente en C mediante el comando:

```
gcc *.c -o MM_ejecutable
```

- Asignación de permisos de ejecución al script en Perl utilizando:

```
chmod +x lanza.pl
```

- Ejecución del script automatizado:

```
./lanza.pl
```

Este procedimiento permitió compilar el código fuente y ejecutar las pruebas de rendimiento de forma automatizada.

3. Diseño de los Experimentos

Para llevar a cabo la evaluación del rendimiento, se definió una batería de pruebas basada en diferentes combinaciones de tamaños de matrices y cantidades de hilos utilizados durante la ejecución. Los parámetros fueron seleccionados con el objetivo de someter al sistema a distintas cargas de trabajo y analizar su comportamiento.

- **Tamaño de la matriz (tamMatriz):** Se utilizaron matrices cuadradas de dimensiones $N \times N$, con valores de N iguales a 1000, 1200, 1400, 1600, 1800, 2000, 2400 y 2800. Estas dimensiones fueron escogidas por su capacidad para generar una carga significativa de procesamiento, permitiendo así evaluar con mayor precisión el impacto del tamaño de entrada en el rendimiento.
- **Número de hilos (NumHilos):** Las pruebas se ejecutaron con 1, 2 y 4 hilos. Esta configuración permitió comparar el comportamiento del algoritmo en modo secuencial (1 hilo) frente a ejecuciones paralelas (2 y 4 hilos), con el fin de observar los beneficios de la paralelización.

4. Automatización de Experimentos

La ejecución de las pruebas se automatizó mediante el uso del script `lanzador.pl`, desarrollado en lenguaje Perl. Este script permitió ejecutar de manera sistemática todas las combinaciones definidas en la batería de pruebas, generando archivos con extensión `.dat` que contienen los resultados correspondientes a cada configuración experimental. Posteriormente, estos datos fueron exportados a una hoja de cálculo para su análisis. Para mayor detalle, el contenido completo del script se encuentra en el archivo `lanza.pl`, incluido en los anexos de este informe.

5. Métricas de Desempeño

La métrica principal utilizada para evaluar el rendimiento fue el tiempo de ejecución, medido en microsegundos. Esta métrica permite cuantificar el tiempo promedio requerido por el sistema para completar la operación de multiplicación de matrices bajo diferentes configuraciones. Para minimizar el efecto de posibles fluctuaciones del sistema operativo, se realizaron 30 repeticiones por configuración, y se utilizó el promedio de estos tiempos como indicador representativo del desempeño.

6. Especificaciones de Hardware y Software

Hardware:

- **Arquitectura:** x86_64
- **CPU(s):** 4 *On-line CPU(s)* list: 0-3
- **Modelo de procesador:** Intel(R) Xeon(R) Gold 6240R CPU @ 2.40GHz
- **Hilos por núcleo:** 1
- **Núcleos por socket:** 1
- **Sockets:** 4
- **Características de virtualización:**
 - **Proveedor de hipervisor:** VMware
- **Jerarquía de memoria:**
 - L1d: 128 KiB (4 instancias)
 - L1i: 128 KiB (4 instancias)
 - L2: 4 MiB (4 instancias)
 - L3: 143 MiB (4 instancias)

Software: En cuanto a las especificaciones de software para la ejecución del taller de multiplicación de matrices, el entorno y las herramientas utilizadas juegan un papel crucial en el rendimiento del programa, especialmente cuando se trabaja en un entorno virtualizado.

Dado que el sistema operativo utilizado es Linux, varias características específicas de este sistema operativo y su ecosistema de herramientas influyen en el rendimiento:

1. **Sistema Operativo Linux:** Linux es reconocido por su capacidad para manejar aplicaciones de alto rendimiento de manera eficiente, y es frecuentemente utilizado en entornos de cómputo intensivo y virtualización. Algunas de las características de Linux que impactan el rendimiento en el contexto de este taller son:
 - **Gestión de Procesos y Hilos:** Linux cuenta con un sistema robusto para manejar múltiples hilos, lo cual es esencial para aplicaciones paralelas como la que se está ejecutando. La creación y sincronización de hilos utilizando `pthread` es eficiente y está optimizada para hardware multinúcleo, permitiendo que cada hilo se ejecute en un núcleo separado cuando es posible.
 - **Planificación de Hilos:** Linux utiliza un planificador de hilos basado en prioridades, lo que permite optimizar la asignación de hilos a los núcleos físicos. En este sistema Linux, que opera con 4 núcleos físicos y 4 hilos de ejecución, el sistema operativo es capaz de distribuir el trabajo de forma balanceada y minimizar los conflictos en el acceso a la CPU y la memoria.

- **Soporte para Virtualización:** Linux se adapta bien a entornos virtualizados y proporciona compatibilidad con hipervisores como VMware, lo que permite que aplicaciones intensivas en CPU, como la multiplicación de matrices, se ejecuten de manera eficiente en máquinas virtuales. Sin embargo, en un entorno virtualizado, el rendimiento puede verse ligeramente afectado por la sobrecarga del hipervisor, aunque, en la práctica, para cálculos paralelos sobre 4 núcleos, este impacto es mínimo.

7. Conclusiones

El taller dedicado a la evaluación del rendimiento permitió estudiar y comparar el comportamiento de un algoritmo de multiplicación de matrices en distintas configuraciones de ejecución, aprovechando la capacidad de paralelización de tres máquinas virtuales con sistema operativo Linux. A través de pruebas tanto en modo secuencial como paralelo, y utilizando configuraciones con 1, 2 y 4 hilos, se pudo observar que el procesamiento paralelo reduce de manera significativa el tiempo de ejecución, particularmente cuando se trabajan con matrices grandes, que requieren una mayor carga computacional. Los resultados evidencian que la elección del sistema de cómputo y la configuración de hilos son elementos cruciales para optimizar el rendimiento.

Además, el uso de herramientas de automatización, como el script `lanzador.pl`, permitió la recolección de datos de forma eficiente y representativa.

Por otro lado, la implementación paralela del algoritmo de multiplicación de matrices, optimizada con la biblioteca `pthread` y ejecutada en un entorno Linux, demostró cómo los sistemas con múltiples procesadores pueden ser aprovechados de manera eficaz para mejorar el rendimiento en aplicaciones que requieren altos recursos computacionales.