Pontificia Universidad Javeriana

2024

Taller De Evaluación de Rendimiento

Integrantes:

David Rodríguez

Juan David Ordoñez Jimenez

Juan Martin Trejos

Sebastián Angúlo

Materia:

Sistemas Operativos

Índice

1. Introducción:

El presente informe describe el taller de evaluación de rendimiento, este taller tiene como propósito principal la evaluación del rendimiento de un algoritmo que realiza multiplicación de matrices, ejecutados en serie y en paralelo, y su comparación en distintos sistemas de cómputo. Con este trabajo, se pretende comprender mejor el impacto de la paralelización en el rendimiento computacional, así como evaluar cómo diferentes configuraciones de hardware y software pueden influir en el tiempo de ejecución de tareas intensivas en cómputo.

En este taller se evaluará el rendimiento de un algoritmo clave: el algoritmo para la multiplicación de matrices (utilizando el algoritmo clásico). Este algoritmo es altamente dependiente de la CPU y demandante en cuanto al procesamiento de datos en paralelo, lo que lo hace ideal para evaluar el impacto de la paralelización en sistemas de cómputo modernos.

El algoritmo de multiplicación de matrices clásico dado para el desarrollo del taller, toma dos matrices A y B, y calcula la matriz C tal que cada elemento C[i][j] es el producto escalar de la fila i de A y la columna j de B. Este proceso es computacionalmente intensivo y se beneficia de la paralelización, ya que cada elemento en C puede calcularse independientemente.

En este algoritmo, el rendimiento se evaluará utilizando 1, 2 y 4 hilos. Con un solo hilo, se espera que el algoritmo funcione en modo secuencial, proporcionando un punto de referencia para la comparación con ejecuciones paralelas. Al utilizar 2 y 4 hilos, el tiempo de ejecución debería reducirse proporcionalmente, aunque los beneficios pueden variar dependiendo de la estructura de cada algoritmo y del overhead de la administración de hilos. En general, se espera que el uso de 4 hilos sea más eficiente que el de 2 en matrices grandes, ya que permite dividir las operaciones en tareas más pequeñas, maximizando la utilización del procesador.

Mas adelante se analizarán y presentaran los resultados obtenidos y se podrá concluir si efectivamente el uso de 2 y 4 hilos reduce el tiempo de ejecución.

2. Descripción del Taller

2.1 objetivos de la Evaluación

Los objetivos de la evaluación se centran en observar cómo la paralelización de un algoritmo puede optimizar el tiempo de ejecución en diferentes plataformas de hardware. La comparación entre ejecuciones en serie y paralelo permitirá identificar

mejoras en el desempeño y validar cómo la configuración del sistema influye en el rendimiento final del algoritmo.

2.2 Actividades previas

Antes de iniciar la evaluación, se realizaron las siguientes actividades:

- Configuración del entorno de trabajo en Linux, ya sea en sistemas nativos o en máquinas virtuales. **En este caso se usaron 3 máquinas virtuales.**
- Descompresión de los archivos fuente provistos.
- Documentación detallada de cada función en los archivos fuente.
- Separación del código en diferentes archivos para organizar la biblioteca de funciones, la interfaz y el archivo principal.
- Análisis y documentación del archivo lanzador.pl (script en Perl), que se utilizó para automatizar la ejecución del programa en diversas configuraciones, que es crucial para la replicación del ejercicio.

2.3 Compilación y ejecución

Para compilar los archivos fuente se utilizaron los siguientes comandos después de comentar y tener listo el lanza.pl

- **gcc *.c -o MM_ejecutable**
- **chmod +x lanza.pl //para convertir el archivo perl en un ejecutable**
- **./lanza.pl**

2.4 Especificaciones de hardware y software

La información respecto al hardware fue obtenida por medio de las maquinas virtuales, ejecutando el comando ‘lscpu’

Hardware:

Arquitectura: x86_64

CPU(s): 4 On-line CPU(s) list: 0-3

Model name: Intel(R) Xeon(R) Gold 6240R CPU @ 2.40GHz

Thread(s) per core: 1

Core(s) per socket: 1

Socket(s): 4

Virtualization features:

Hypervisor vendor: VMware

Jerarquía de Memoria:

L1d: 128 KiB (4 instances)

L1i: 128 KiB (4 instances)

L2: 4 MiB (4 instances)

L3: 143 MiB (4 instances)

- Mas adelante se hablara de las consideraciones respecto estas especificaciones

Software:

En cuanto a las **especificaciones de software** para la ejecución del taller de multiplicación de matrices, el entorno y las herramientas utilizadas también juegan un papel importante en el rendimiento del programa, especialmente al trabajar en un entorno virtualizado.

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

1. Sistema Operativo Linux

Linux es conocido por su capacidad para manejar aplicaciones de cómputo intensivo de manera eficiente y es comúnmente utilizado en entornos de alto rendimiento y virtualización. Algunas características de Linux que afectan el rendimiento en el contexto de este taller incluyen:

- **Gestión de Procesos y Hilos**: Linux tiene un sistema robusto para manejar múltiples hilos, lo cual es clave para aplicaciones paralelas como esta. La creación y sincronización de hilos con pthread (biblioteca POSIX de Linux para manejo de hilos) es eficiente en Linux y está optimizada para hardware multi-core, permitiendo que cada hilo se ejecute en un núcleo separado siempre que sea posible.

- **Planificación de Hilos**: Linux utiliza un planificador de hilos basado en prioridades y puede optimizar la asignación de hilos a núcleos físicos. Dado que el sistema Linux en esta máquina virtual opera con 4 núcleos físicos y 4 hilos de ejecución, el sistema operativo tiene una gran capacidad para distribuir el trabajo de manera balanceada y minimizar conflictos en el acceso a CPU y memoria.
- **Soporte para Virtualización**: Linux se adapta bien a entornos virtualizados y proporciona una capa de compatibilidad con hypervisores como VMware, que permite que aplicaciones intensivas en CPU, como la multiplicación de matrices, se ejecuten de manera efectiva en máquinas virtuales. Sin embargo, en una VM, el rendimiento puede verse afectado ligeramente por el overhead del hypervisor, aunque en la práctica, para cálculos en paralelo sobre 4 núcleos, el impacto es mínimo.

2. Bibliotecas de Hilos POSIX (Pthreads)

El programa utiliza pthread para la creación y manejo de hilos, lo cual es ideal en Linux dado su soporte nativo para POSIX:

- Facilidad de Manejo de Hilos: La biblioteca pthread permite un control detallado sobre los hilos (creación, sincronización y finalización) y proporciona mecanismos de sincronización, como mutexes (pthread_mutex_t) para evitar condiciones de carrera, lo cual es esencial en operaciones de cómputo intensivo como la multiplicación de matrices.
- Compatibilidad con el Sistema Multi-core: En combinación con Linux, pthread permite que cada hilo corra en paralelo en diferentes núcleos, maximizando la utilización de CPU y distribuyendo eficientemente el trabajo.

3. Metodología de Evaluación

3.1 Batería de Experimentación

Para la experimentación, se definieron varias combinaciones de tamaños de matriz y números de hilos.

- **Tamaño de la Matriz (tamMatriz):** Se eligieron diferentes dimensiones de matrices para observar cómo influye el tamaño en el rendimiento, Se usaron

los siguientes tamaños de NxN para el experimento:
1000,1200,1400,1600,1800,2000,2400,2800.
Estos tamaños fueron escogidos por que permiten de manera efectiva medir el rendimiento y exigir al procesador.

- **Número de Hilos (NumHilos):** Las pruebas se realizaron con diferentes números de hilos (1, 2, 4), para comparar ejecuciones en serie (1 hilo) y en paralelo.

3.2 Automatización de Experimentos

Para automatizar la ejecución de las pruebas se empleó el script lanzador.pl, el cual permite ejecutar la batería de experimentos de forma automatizada. Este script escrito en lenguaje Perl, genera archivos .dat con los resultados de cada configuración de prueba, que luego se exportaron a una hoja de cálculo para el análisis. Los demás detalles se encuentran en el archivo lanza.pl anexado a este informe.

3.3 Métricas de Desempeño

Para evaluar el desempeño, se utilizó el tiempo de ejecución, medido en microsegundos como métrica principal. Esta métrica permite medir el tiempo promedio que toma cada configuración para completar el cálculo de la multiplicación de matrices, considerando la variabilidad introducida por el sistema operativo. Se tuvo en cuenta el promedio de los tiempos, calculado tras 30 repeticiones, que nos ofrece una estimación representativa del rendimiento.

4. Resultados y Análisis

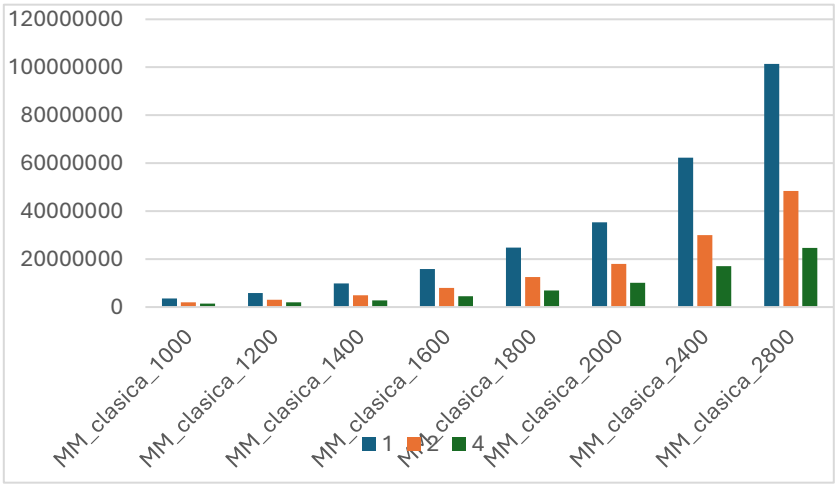
4.1 Tabla de Resultados

En las siguientes tablas solo se mostrarán los promedios obtenidos después de la extracción de todos los datos en cada equipo, las demás tablas con todos los datos obtenidos serán anexadas a este informe.

	Hilos		
Tamaños	1	2	4
MM_clasica_1000	3545676	2011252	1428958
MM_clasica_1200	5822049	3085136	1977128
MM_clasica_1400	9879597	4918215	2811473
MM_clasica_1600	15867966	7973380	4472628
MM_clasica_1800	24746263	12499045	6902627

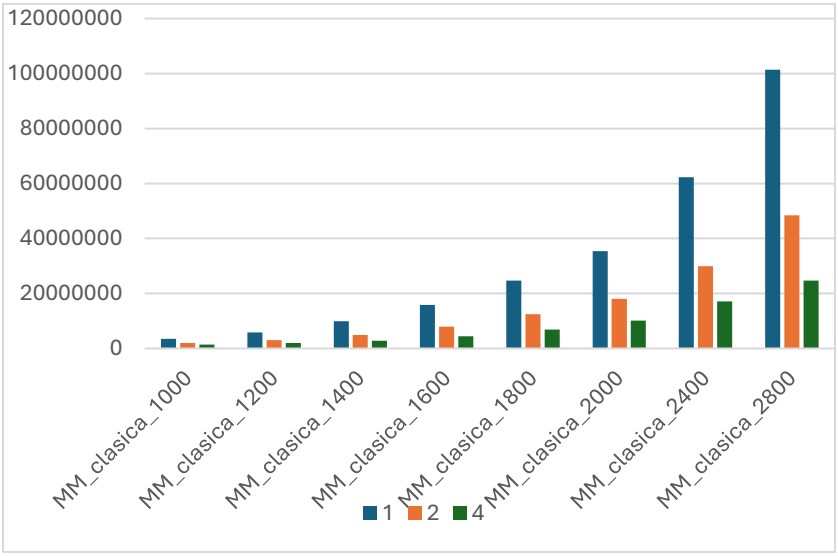
MM_clasica_2000	35366649	18058996	10121552
MM_clasica_2400	62246624	29955882	17090424
MM_clasica_2800	101412705	48389483	24730128

1. Tabla de Promedios Equipo 1



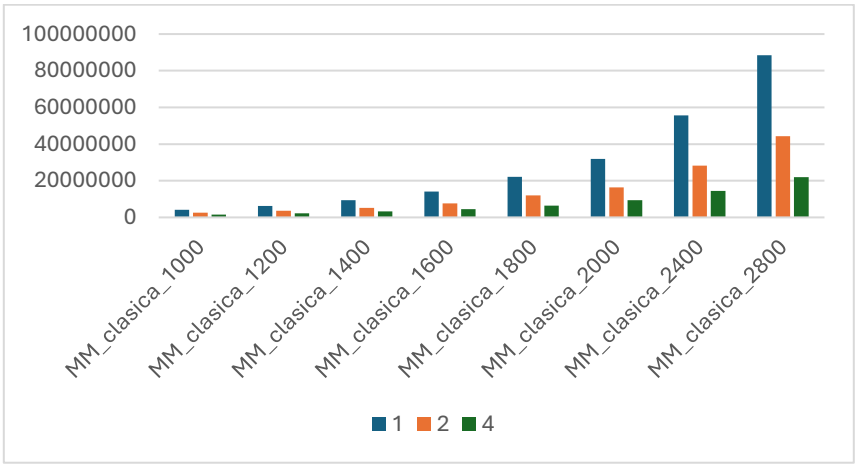
	Hilos		
Tamaños	1	2	4
MM_clasica_1000	3430471	1935851	1341036
MM_clasica_1200	5465113	2962496	1890195
MM_clasica_1400	8606520	4604873	2780290
MM_clasica_1600	14011773	7096700	4050952
MM_clasica_1800	22589771	11632728	6221276
MM_clasica_2000	33585521	17344887	9778826
MM_clasica_2400	59447370	28579489	15993110
MM_clasica_2800	101202286	50400596	28347949

2. Tabla de Promedios Equipo 2



	Hilos		
Tamaños	1	2	4
MM_clasica_1000	4101121	2489848	1449910
MM_clasica_1200	6205671	3662767	2262104
MM_clasica_1400	9390394	5221105	3272360
MM_clasica_1600	14162461	7703787	4551595
MM_clasica_1800	22082272	11952374	6422215
MM_clasica_2000	31893754	16433239	9416613
MM_clasica_2400	55584765	28232302	14501001
MM_clasica_2800	88508751	44226960	21901520

3. Tabla de Promedio Equipo 3



4.2 Análisis de resultados:

- **Consideraciones del hardware y de la máquina virtual en el rendimiento**

Las máquinas virtuales utilizadas cuentan con las siguientes características clave:

- **Arquitectura**: x86_64 con capacidad de operación en modos de 32 y 64 bits.
- **Hilos y núcleos**: La máquina tiene configurados 4 núcleos, cada uno con un hilo. Este número de núcleos limita el máximo de paralelismo a cuatro hilos efectivos.
- **Memoria caché**: L3 de 143 MiB compartida entre los núcleos, además de cachés L1 y L2 individuales. Esto puede ser muy beneficioso, ya que los datos de matrices pueden ser pre-cargados y manipulados en memoria caché, reduciendo la latencia de acceso a memoria.

al tratarse de una CPU con un número significativo de núcleos y una gran cantidad de caché L3, se pudo ver que los cálculos matriciales paralelos se ejecutaron de manera efectiva en cuatro hilos.

- **Comparación Serie vs. Paralelo**

Las ejecuciones en paralelo lograron tiempos de ejecución considerablemente menores en comparación con las ejecuciones en serie, especialmente en matrices grandes, donde a medida que se iba agrandando la matriz la diferencia entre el hilo 1 y los hilos 2 y 4 era cada vez mayor. Sin embargo, en matrices pequeñas, el impacto de la paralelización es menor debido al overhead de gestión de hilos.

- Además, se pudo ver que los resultaron fueron muy parecidos con las 3 maquinas virtuales, estos no fueron iguales ya que aunque las máquinas virtuales tienen configuraciones de hardware casi idénticas, las variaciones en sus entornos de ejecución influyen en los resultados debido a la forma en que cada hypervisor administra y asigna recursos, la carga de trabajo del host, y el manejo de memoria y sincronización de hilos en cada caso. Estas diferencias, aunque pequeñas, pueden afectar el proceso de búsqueda de obtener los resultados mas consistentes y cercanos al rendimiento máximo posible.

5. Conclusiones

El taller de evaluación de rendimiento permitió analizar y comparar el comportamiento de un algoritmo de multiplicación de matrices en diferentes configuraciones de ejecución, aprovechando la capacidad de paralelización de 3 máquinas virtuales con sistema operativo Linux. A través de pruebas en serie y en paralelo, y utilizando configuraciones de 1, 2 y 4 hilos, fue posible observar cómo el procesamiento paralelo reduce significativamente el tiempo de ejecución, especialmente en el caso de matrices grandes que presentan una mayor carga computacional. Los resultados muestran que la elección del sistema de cómputo y la configuración de hilos son factores determinantes en el rendimiento. Además, el uso de herramientas de automatización, como el script lanzador.pl, facilitó la recolección de datos de manera eficiente y representativa.

Además, La implementación paralela del algoritmo de multiplicación de matrices, optimizada mediante la biblioteca pthread y ejecutada en un entorno Linux, demostró cómo los sistemas multi-procesador pueden ser utilizados eficazmente para mejorar el rendimiento en aplicaciones que necesitan altos recursos computacionales.

Bibliografías:

1. Torvalds, L., & Hamano, J. (2023). *The Linux Kernel Archives*. Recuperado de https://www.kernel.org

2. Institute of Electrical and Electronics Engineers. (2017). *IEEE Standard for Information Technology – Portable Operating System Interface (POSIX) Base Specifications, Issue 7*. IEEE. https://doi.org/10.1109/IEEESTD.2018.8277153