SISTEMAS OPERATIVOS

TALLER MEDICIONES

PROFESOR:

John Corredor

INTEGRANTES:

Mateo Maldonado

Edward Quintero

Juan Esteban Garzón

Felipe Garrido Flores

08 de noviembre del 2024

1. Introducción:

En el presente documento se analizará y evaluará el rendimiento de un algoritmo de multiplicación de matrices, tanto en su versión clásica como transpuesta.

La multiplicación de matrices es una operación fundamental para muchas aplicaciones, y su eficiencia puede tener un impacto significativo en el rendimiento global de estos sistemas. Para este taller se utilizó la implementación del algoritmo clásico, utilizando diferentes números de hilos en ejecución. Para poder evaluar al algoritmo se utilizarán cuatro sistemas distintos: equipo propio real, máquinas virtuales, un sistema en la plataforma Cocalc y Replit. Se medirán tiempos de ejecución del algoritmo para matrices de diferentes tamaños con diferentes números de hilos.

En este taller previamente se tuvo que obtener conocimientos previos sobre entendimiento de archivos fuente en C y Perl, la configuración de Perl para automatizar la ejecución y captura de datos. Previo a esto se realizarán distintas pruebas ejecutando el programa con varios tamaños de matriz, hilos y se analizarán dichos resultados.

2. Métricas de desempeño:

Para evaluar el desempeño de los programas de multiplicación de matrices, se realizará una comparación, en cómo actúa este algoritmo en distintas máquinas, configuraciones y sistemas. Estas métricas nos permiten obtener una visión clara y cuantitativa del rendimiento del algoritmo en diferentes condiciones y configuraciones. A continuación, se describen las métricas de desempeño utilizadas:

- **Tiempo de ejecución:** es la métrica principal utilizada, mide el tiempo total que tarda el algoritmo en completar la multiplicación de dos matrices. El tiempo de ejecución se mide en microsegundos.

  Fórmula:

  $$T(ejecución) = T(fin) - T(inicio)$$

  Donde:

  - T(inicio) es el tiempo en que comienza la ejecución del algoritmo.
  - T(fin) es el tiempo en que finaliza la ejecución del algoritmo.

- **Speedup:** El speedup mide la mejora en el tiempo de ejecución cuando un algoritmo paralelo se ejecuta en múltiples hilos en comparación con la ejecución secuencial (un solo hilo).

Formula:

$$Speedup = \frac{T(serie)}{T(paralelo)}$$

Donde:

- T(serie) es el tiempo de ejecución del algoritmo en un solo hilo.
- T(paralelo) es el tiempo de ejecución del algoritmo utilizando múltiples hilos.

- **Tiempo de ejecución promedio:** Dado que los tiempos de ejecución pueden variar debido a factores del sistema operativo y otros elementos, se realizarán múltiples ejecuciones (al menos 30) para cada configuración y se calculará el tiempo de ejecución promedio para obtener un valor representativo y confiable.

Fórmula:

$$T(promedio) = \frac{\sum_{i=1}^{N} T_i}{N}$$

Donde:

- N es el número de ejecuciones.
- T_i Es el tiempo de ejecución de la (i)-ésima ejecución.

3. **Descripción Hardware y Software en los sistemas probados:**
   - Sistema 1 – Dispositivo local propio
     - **Procesador:** Intel Core i7
     - **Memoria:** 16 GB RAM
     - **Sistema Operativo:** Terminal Ubuntu 20.04

   - Sistema 2 – Máquina Virtual
     - **Procesador:** Intel Xeon
     - **Memoria:**
       - L1d: 192 KiB (4 instances)
       - L1i: 128 KiB (4 instances)
       - L2: 5 MiB (4 instances)
       - L3: 168 MiB (4 instance)
     - **Sistema Operativo:** Rocky Linux 9.4

   - Sistema 3 – Sistema de Cocalc
     - **Procesador:** Intel Xeon

    - **Memoria:**
        - L1d: 64 KiB (2 instances)
        - L1i: 64 KiB (2 instances)
        - L2: 2 MiB (2 instances)
        - L3: 33 MiB (1 instance)
- **Sistema Operativo:** Ubuntu 22.4

- Sistema 4 – Sistema de Replit
    - **Procesador:** Intel Xeon
    - **Memoria:**
        - L1d: 128 KiB
        - L1i: 128 KiB
        - L2: 2 MiB
        - L3: 32 MiB
- **Sistema Operativo:** Ubuntu 20.04

4. **Cambios en los archivos:**

4. 1 Modificaciones que se realizaron al archivo "Fuente_Evaluacion.c"

- Se añadió documentación detallada en forma de comentarios, para cada funcionalidad que realiza el programa esto para más entendimiento.
- Se añadió una variable double elapsed_time para almacenar el tiempo total de ejecución en microsegundos.
- La finalización del tiempo se consolidó con la operación de resta entre stop y start, almacenando el resultado en elapsed_time.
- Se modificó la salida que imprimía al ejecutarse el programa incluyendo una salida en formato CSV al finalizar la ejecución.
- Se reorganizó el código para que sea más claro y legible, agrupando las inicializaciones y explicando los pasos en el bloque main.

4.2 Las modificaciones que se realizaron en el archivo de "lanza.pl"

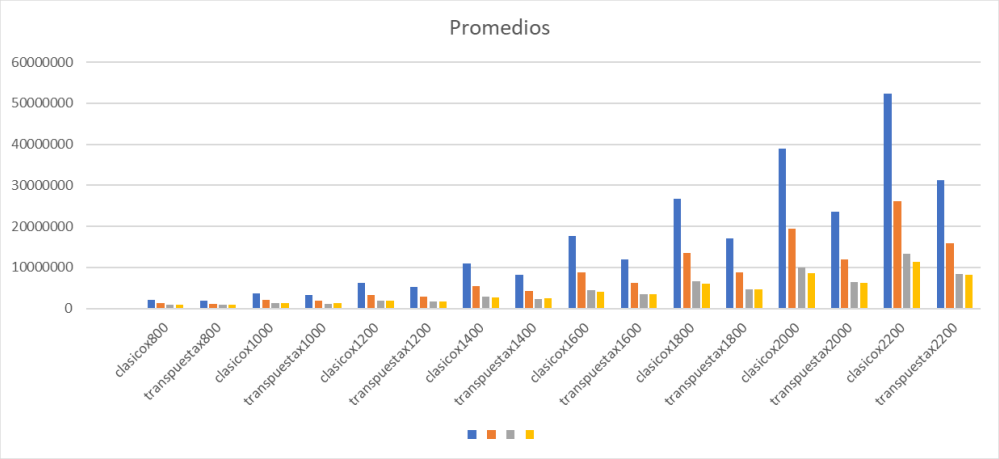
- Se añadieron comentarios para describir la función de cada bloque principal. Los encabezados explican la inicialización de variables, el bucle de ejecución de pruebas y el procesamiento de resultados.
- Se añade y explica el proceso de abrir el archivo, el formato de la salida esperada en CSV, y cómo se manejan los errores de formato.
- Al principio, se añadieron use strict; y use warnings, esto para ayudar a detectar errores y advertencias en tiempo de ejecución.

- Se añadieron tamaños de matriz y número de hilos. En la lista @Size_Matriz se añadieron valores adicionales (800, 1000, 1200, 1400, 1600, 1800, 2000, 2200), y en @Num_Hilos se añadió 4 (1, 2, 4, 8) , lo que permite una mejor ejecución de las pruebas.
- Se crea un archivo en formato CSV con el fin de almacenar todas las pruebas realizadas.
- Se modificó para que pueda capturar la salida de cada ejecución y extrae los valores con una expresión regular (if ($resultado =~ /^(\d+),(\d+),(\d+)/)), validando que el formato de salida sea correcto (tamaño, hilos, tiempo en microsegundos).
- Se le agregó funcionalidad para acumular el tiempo total, para ir sumando el tiempo de ejecución actual.
- Se añade función para que calcule el promedio de ejecuciones realizadas y lo muestra en la terminal

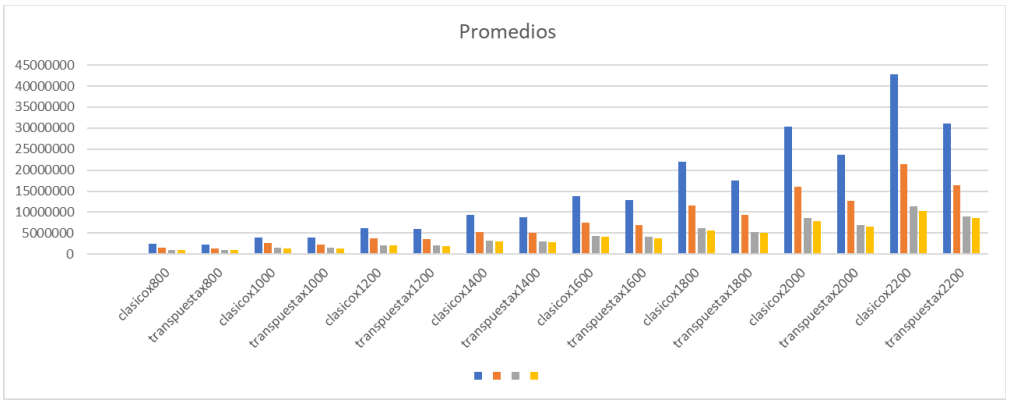
5. **Gráficas:**

5. 1 Diferentes gráficas obtenidas en la Máquinas Virtuales.

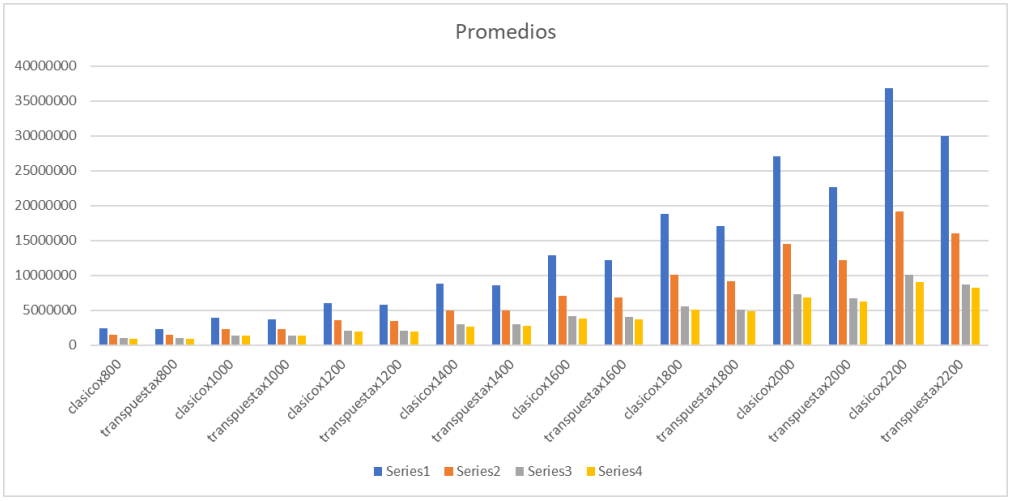
- Máquina Virtual de Mateo Maldonado:



- Máquina Virtual de Edward Quintero:

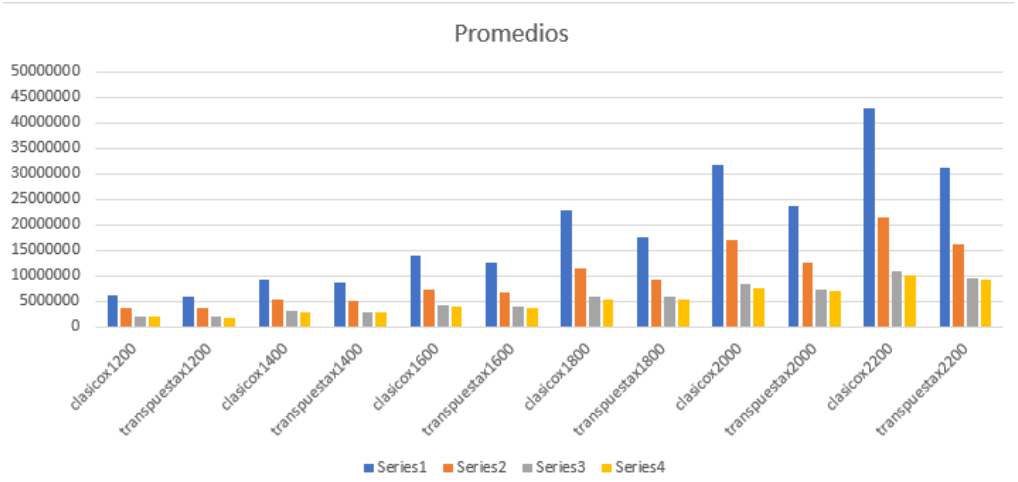


- Máquina Virtual de Felipe Garrido:

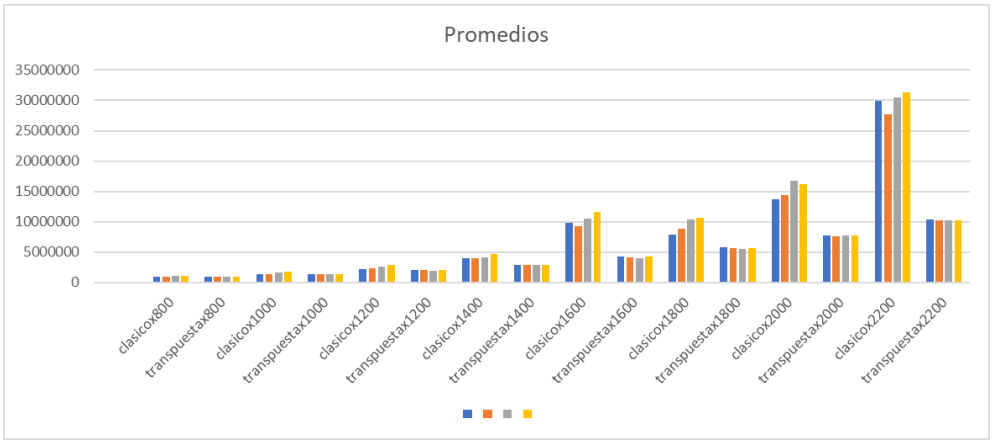


5. 2 Diferentes gráficas obtenidas en los Equipos Personales.

- Equipo personal de Juan Garzón

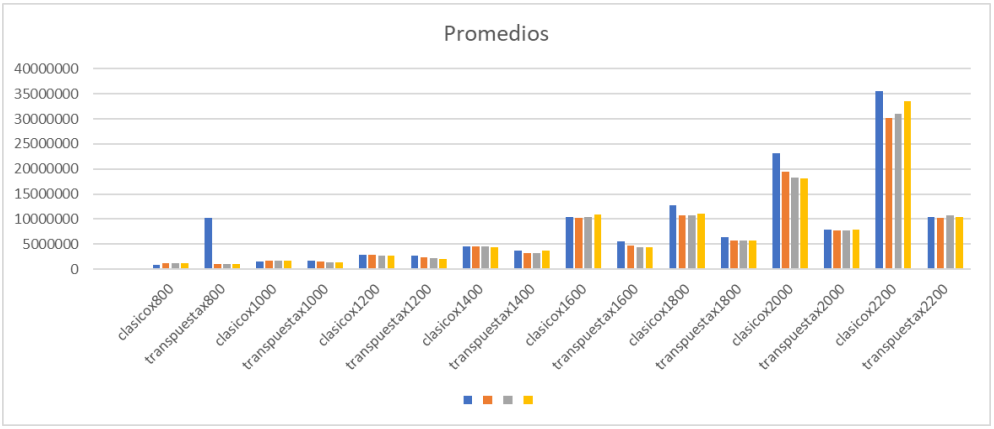


- Equipo personal de Mateo Maldonado:



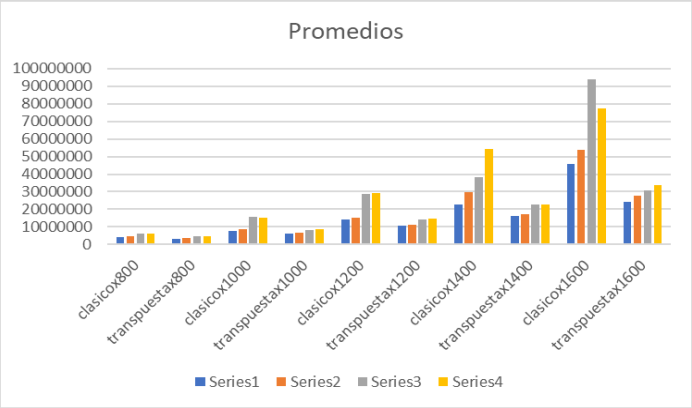
5. 2 Gráfica obtenida por Replit.

- Gráfica obtenida en Replit de Edward Quintero:



5. 3 Gráfica obtenida por Cocalc.

- Gráfica obtenida en Cocalc de Juan Garzón:



6. **Análisis de datos:**

Al analizar los datos obtenidos y las gráficas modeladas, es importante resaltar la relación entre el tiempo de ejecución y la cantidad de hilos, obteniendo diversos resultados al probar en dos entornos distintos, se obtienen las siguientes observaciones.

Observaciones:

- Tenemos una tendencia general, en la que observamos que mientras mayor sea el número de hilos presentes, se genera una disminución en el tiempo de ejecución, lo que nos lleva a concluir que el paralelismo mejora el rendimiento de nuestro algoritmo.
- Teniendo en cuenta los diferentes equipos utilizados podemos observar que los equipos con mejor hardware, obtienen una gran mejora en la utilización de hilos, mientras que en Replit y Cocalc no lo hace por culpa de su hardware, como pudimos observar Cocalc no pudo realizar por completo todas las multiplicaciones de matrices.
- Algo interesante que se observa es que en algunos casos mientras mayor sea el número de hilos, el tiempo puede aumentar, nos lleva a la conclusión de que a la hora de escoger hilos es importante tener en cuenta el rendimiento del equipo y las características que posee.
- Para aprovechar al máximo el rendimiento en la multiplicación de matrices, el uso del algoritmo transpuesto combinado con varios hilos es la mejor opción, especialmente en matrices de tamaño mediano a grande.

7. **Conclusiones:**

Las pruebas que se realizaron nos mostraron que agregando más hilos mejora mucho el procesamiento del algoritmo de multiplicación de matrices, pero para esto toca tener muy en cuenta cuales son los recursos dentro de la máquina la cual se va a utilizar. En los

sistemas con más recursos que se utilizaron como el equipo personal y la máquina virtual, se notó más la diferencia de usar más hilos que en los otros sistemas, por lo que no tienen un hardware de buen rendimiento, lo que hizo esto fue que el algoritmo no fuera tan rápido.

También la automatización a la hora de recolectar datos nos ayudó a obtener datos de los que nos podemos fiar, eliminamos datos “ruidosos” que interfirieran en los resultados para que fueran más exactos. Gracias a esto se pudo observar detalladamente la ayuda de los hilos.