



UNIVERSIDAD
SERGIO ARBOLEDA

Primer Parcial: Comparativa de Rendimiento 4 Máquinas - 2 Benchmarks

Luis Alejandro Vejarano Gutierrez
Escuela de Ciencias Exactas e Ingeniería
Universidad Sergio Arboleda -Bogotá, Colombia
luis.vejarano01@correo.usa.edu.co

Jessica Valentina Parrado Alfonso
Escuela de Ciencias Exactas e Ingeniería
Universidad Sergio Arboleda -Bogotá, Colombia
jessica.parrado01@correo.usa.edu.co

Lady Geraldine Salazar Bayona
Escuela de Ciencias Exactas e Ingeniería
Universidad Sergio Arboleda -Bogotá, Colombia
lady.salazar01@correo.usa.edu.co

Michael Steven Pinilla Mateus
Escuela de Ciencias Exactas e Ingeniería
Universidad Sergio Arboleda -Bogotá, Colombia
michael.pinilla01@correo.usa.edu.co

Computación Paralela
Marzo 2021

Resumen: Se busca comparar tiempos de ejecución de dos programas (ordinario y con hilos) en donde se realiza una multiplicación de 2 matrices de manera eficiente. Para esto, se propone el uso de hilos y módulos para la resolución rápida y eficaz del problema; se plantea solucionarlo con la comparación de procesos y el tiempo de ejecución de cada uno de ellos para llegar a la mejor opción usando perl, makefile y python. Por lo que se espera obtener mejores resultados de tiempo de ejecución en el programa basado en hilos que en la multiplicación ordinaria, ya que este con la creación de subprocesos ayuda a mejorar el rendimiento.

Palabras Claves: Multiplicación, Módulos, Hilos, Makefile, Perl, Python, C y Matrices.

I. INTRODUCCIÓN

El mundo de la programación está constituido principalmente por matrices, las cuales son fundamentales para los procesos y algoritmos que se desarrollan y utilizan hoy en día. Es de resaltar que en este curso estamos profundizando en cómo optimizar procesos y memoria de los equipos, utilizando al máximo todos los recursos, por eso es fundamental saber cómo funcionan y desarrollan los módulos y cómo se deben manejar las matrices en estos para sacar el mejor provecho.

La multiplicación de matrices aunque es un tema simple, es esencial en la creación de programas y aplicaciones los cuales pueden ayudar en la resolución de los problemas del mundo real.

II. METODOLOGÍA

La intención de este proyecto es poder analizar cómo el uso de los hilos, módulos y procesos permiten

maximizar la eficacia de los programas y optimizar los procesos de la computadora, reduciendo el tiempo de respuesta y distribuyendo la carga de trabajo en cada procesador.

Para analizar cómo trabaja el programa pthread, cuya multiplicación se repartía en hilos, se ejecutó en los diferentes computadores para tratar los datos y comparar dependiendo de la cantidad de procesadores y el sistema operativo.

Configuración de las máquinas:

1) Máquina 1

Número de núcleos: 4

RAM (GB): 4

Sistema Operativo: Windows 10 de 64 bits

Procesador: AMD A9-9425 Radeon R5 x64

Velocidad del procesador: 3.10 GHz

2) Máquina 2

Número de núcleos: 4

RAM (GB): 8

Sistema Operativo: Windows 10 de 64 bits
ejecutado por medio de terminal de ubuntu

Procesador: Intel(R) Core(TM) i5-6200U

Velocidad del procesador: 2.30 GHz

3) Máquina 3

Número de núcleos: 8

RAM (GB): 4

Sistema Operativo: Windows 10 de 64 bits
ejecutado por medio de terminal de ubuntu.

Procesador: Intel(R) Core(TM) i5-8250U

Velocidad del procesador: 1.60 GHz

4) Máquina 4

Número de núcleos: 8

RAM (GB): 16

Sistema Operativo: Windows 10 de 64 bits
ejecutado por medio de terminal de ubuntu.

Procesador: Procesador Intel(R) Core(TM)
i5-8300H

Velocidad del procesador: 2.30 GHz

Para obtener los resultados se utilizó un programa en Perl para que el sistema se ejecute 30 veces por cada configuración diferente, para luego almacenar el tiempo de ejecución en segundos en archivos. Es por eso que fue necesaria la creación de un script que automatice en Python la lectura de cada archivo y calcule el promedio de cada ejecución.

III. RESULTADOS

Luego de realizar las 30 ejecuciones (cantidad mínima para obtener un promedio correcto) de cada uno de los benchmarks, para cada tamaño y número de hilos (si es el caso), se crean las siguientes gráficas que muestran el tiempo (segundos) que dura hacer cada multiplicación de matrices para cada máquina:

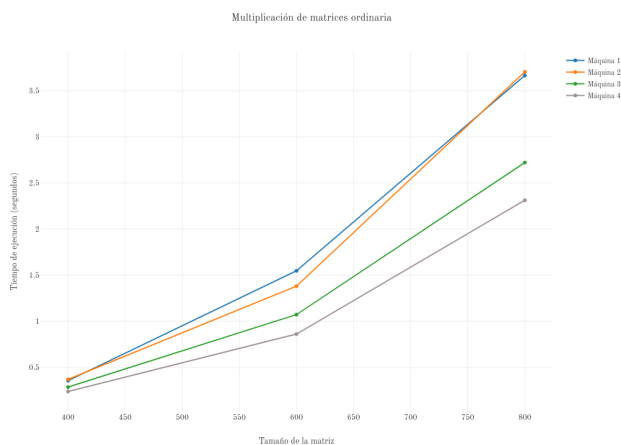


Fig 1. Gráfica del rendimiento de cada equipo para la multiplicación ordinaria.

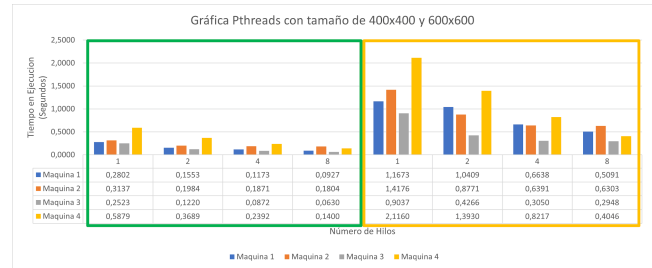


Fig 2. Gráfica del rendimiento de cada equipo usando 1, 2, 4 y 8 hilos para matrices de 400 por 400 y 600 por 600.

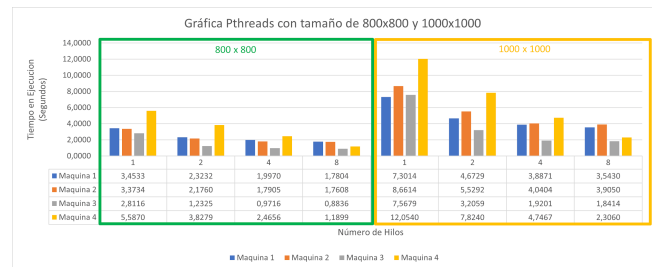


Fig 3. Gráfica del rendimiento de cada equipo usando 1, 2, 4 y 8 hilos para matrices de 800 por 800 y 1000 por 1000

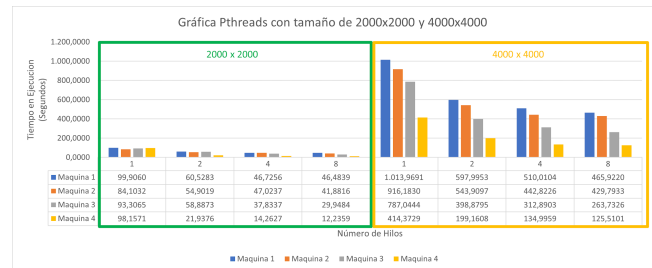


Fig 4. Gráfica del rendimiento de cada equipo usando 1, 2, 4 y 8 hilos para matrices de 2000 por 2000 y 8000 por 8000.

Teniendo en cuenta las gráficas se demuestra que en la ejecución de las matrices del programa “Ordinaria” (sin hilos) la máquina 4 la cual tiene los mejores componentes. Sin embargo, para la máquina 2 y máquina 1, el rendimiento fue muy bajo ya que tuvo tiempos muy altos, debe ser porque sus componentes no son tan eficientes como la otra máquina.

Por otra parte, en la sección de las gráficas acerca de la división de hilos, se puede observar que la máquina 4 la que debería tener mejor rendimiento, tuvo los peores tiempos en las matrices de 400x400, 600x600, 800x800 y 1000x1000, se puede decir que puede ser porque pudo haber tenido programas en segundo plano (programas que se ejecutan por defecto) que mancharon los datos. En cambio, en las matrices de 2000 x 2000 y 4000 x 4000 tuvo los mejores tiempos, usando de manera eficiente todos sus componentes.

Así mismo, para las matrices de 400x400, 600x600, 800x800 y 1000x1000 la máquina 3 tuvo el mejor

rendimiento y para las matrices 2000 x 2000 y 4000 x 4000 la máquina 1 tuvo los peores tiempos.

IV. CONCLUSIONES

Durante el proceso se aprende acerca de la aplicación de módulos y de diferentes hilos para la multiplicación de matrices teniendo en cuenta los tamaños de estas y así comprender de manera práctica, como la utilización de hilos mejora el tiempo de ejecución.

Además, al analizar y comparar las gráficas de los tiempos de ejecución, se puede determinar que la máquina 4 a nivel general tuvo el mejor rendimiento, ya que fue el más rápido. Sin embargo, las máquinas 1 y 2 fueron las más deficientes, teniendo tiempos muy altos. Por otro lado, la máquina 3 se mantuvo equilibrada, teniendo tiempos medios con respecto a las otras máquinas.

También, se ve en las gráficas del programa con pthreads que la máquina 4 tiene un mejor rendimiento en matrices grandes (en las de tamaño 2000 y 4000) en comparación a las más pequeñas. Aunque, para el programa de multiplicación ordinaria se evidencia que esta misma máquina tiene el mejor rendimiento de todas. Existiendo probablemente programas en segundo plano que se hayan ejecutado y así aumentado el tiempo de ejecución “manchando” algunos datos.

Para finalizar, se tuvo que ser muy minuciosos en la ejecución, teniendo en cuenta los requisitos pedidos, para que así la toma de rendimientos cumpliera el objetivo satisfactoriamente.

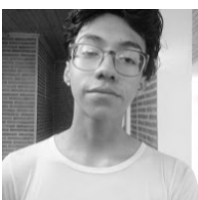
V. REFERENCIAS

[1] Shuo Yang (2014) *Parallel-Programming*. Github. Disponible en: <https://github.com/imsure/parallel-programming/blob/master/matrix-multiplication/matrix-mul-pthread.c>

VI. BIOGRAFÍAS



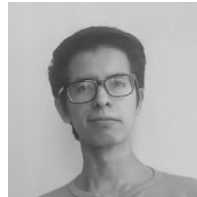
Jessica Valentina Parrado Alfonso nació el 31 de Enero del 2001 en Bogotá, Colombia. Actualmente, es estudiante de Ingeniería de Sistemas y Telecomunicaciones en la Universidad Sergio Arboleda



Michael Steven Pinilla Mateus nació el 9 de Octubre de 2000 en Bogotá, Colombia. En el presente estudia la carrera profesional de Ingeniería de Sistemas y Telecomunicaciones en la Universidad Sergio Arboleda



Lady Geraldine Salazar Bayona nació el 6 de Agosto de 1998 en Bogotá, Colombia. En los últimos años ha estado estudiando Ingeniería de Sistemas y Telecomunicaciones en la Universidad Sergio Arboleda.



Luis Alejandro Vejarano Gutierrez nació el 22 de Mayo del 2000 en Bogotá, Colombia.. Se desempeña como estudiante de Ingeniería de Sistemas en la Universidad Sergio Arboleda.