

Practica 1 - Hilos y procesos

Dorian Abad Díaz, Diego Clemente Rojas
Universidad Nacional Bogotá, Colombia
Email: datovard@unal.edu.co, dicrojasch@unal.edu.co

Resumen—Este documento contiene la explicación detallada de una serie de pruebas realizadas con el fin de cuantificar el rendimiento del procesamiento de un computador. Este rendimiento se medirá principalmente de 3 maneras; una es mediante la ejecución de un programa secuencial, la segunda mediante un programa que implementa varios procesos y la última mediante un programa que implementa hilos. Con el fin de comparar estas 3 implementaciones se realizarán las mismas pruebas, donde cada implementación resolverá el mismo problema y de esta manera se busca identificar y medir sus diferencias. Adicionalmente se realizarán diferentes pruebas donde se aumentará el número de hilos y procesos para analizar sus cambios de rendimiento y de esta manera identificar los límites de rendimiento de estas aplicaciones.

I. INTRODUCCION

En el siguiente documento se describe el funcionamiento de método de la paralelización del problema de multiplicación de dos matrices cuadradas, utilizando dos tecnologías de computación, el primero es el uso de hilos de programa, haciendo uso de la hilos POSIX, y el segundo método es el uso de múltiples procesos de máquina haciendo uso de `fork()`, ambos métodos soportados en el lenguaje de programación C.

Además del funcionamiento, se realiza el análisis del rendimiento de cada uno de éstos métodos de acuerdo a dos factores: speedup, que representa la mejora en la velocidad de ejecución de una tarea cuando se realizan cambios en su arquitectura, mientras que el segundo factor determinante es el response time o tiempo de respuesta, es el tiempo que tarda un programa para reaccionar a una entrada dada.

II. MÉTODOS DE PARALELIZACIÓN

II-A. Utilizando el método de hilos

Para comenzar, necesitamos una serie de parametros con los que trabajar, primero tenemos que conocer la cantidad de hilos que deseamos lanzar en cada ejecución y el tamaño que van a tener las matrices, para esto declaramos las variables `N` e `HILOS`, que determinan el tamaño de la matriz a multiplicar y la cantidad de hilos a utilizar, respectivamente; ahora debemos crear las matrices que necesitamos en nuestra operación $A * B = C$, es por esto que declaramos los apuntadores dobles `A`, `B` y `C`, en los cuales almacenaremos las matrices, la matriz `A` y `B` serán llenadas con números aleatorios.

Al iniciar el proceso principal, lo primero que realizamos es recibir la cantidad de hilos y el tamaño de la matriz, con los

valores recibidos, creamos un arreglo que almacena cada hilo lanzado y otro arreglo que almacena un identificador de cada hilo con el fin de un mejor control de las ejecuciones de los hilos.

El método de particionamiento del problema fue realizado mediante la función llamada *“calculateMul”*. Los hilos creados ejecutan esta función y antes de ejecutarla a cada hilo se asigna un identificador único, este es enviado como argumento en la función. En la función se usa este identificador para asignar a cada hilo una o varias filas de la matriz de respuesta `C`.

Cada hilo podrá realizar las operaciones que se necesitan para producir una fila de la matriz de respuesta `C`, teniendo que multiplicar y sumar la fila asignada de la matriz `A` con cada una de las columnas de la matriz `B`.

En el caso en que tenemos que con una cantidad de hilos `H`, un tamaño de matriz `N` y además tenemos que $H < N$, sucede que los hilos sólo producen las primeras `H` filas de la matriz `C`, pero aún nos hacen falta $N - H$ filas

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & a_{2,3} & \cdots & a_{2,n} \\ & \vdots & & \ddots & \vdots \\ a_{n-h,1} & \cdots & & \ddots & a_{n-h,n} \\ a_{n-h+1,1} & \cdots & & \ddots & a_{n-h+1,n} \\ a_{n,1} & a_{n,2} & a_{n,3} & \cdots & a_{n,n} \\ & \vdots & & \ddots & \vdots \end{pmatrix}$$

Es entonces donde el primer hilo puede calcular la primera fila de `C`, pero también puede ejecutar la fila $N - H + 1$, el mismo caso será para el segundo hilo, que puede calcular la segunda fila y adicionalmente la fila $N - H + 2$ y así sucesivamente.

II-B. Utilizando el método de procesos

Al pasar al método de procesos, podemos notar que aún necesitamos el tamaño de la matriz, ahora no necesitamos una cantidad de hilos, pero si una cantidad de procesos que deseamos ejecutar, entonces necesitaremos las variables `N` y `PROCESOS` en nuestro programa.

A diferencia de los hilos, en nuestro programa de procesos

no poseemos una matriz global C , con el fin de evitar que al realizar la creación de los procesos adicionales, se repitan valores o uso de memoria innecesarios.

Luego de recibir los parámetros que necesitamos, creamos un arreglo que contiene los identificadores de los procesos que creemos un arreglo de arreglos donde almacenaremos pipes para la comunicación interprocesos

Ahora realizaremos la creación de los procesos solicitados, donde cada proceso va a realizar la misma tarea que realizaba cada hilo en el método anterior, con la diferencia de que no puede almacenar los datos en una variable global, sino que debe retornar la fila de la matriz respuesta que calculó y el proceso principal se encarga de reunir los resultados en una única matriz.

III. RESULTADOS DE LAS MEDICIONES

Para las distintas mediciones que realizamos a los métodos, tenemos un límite común del tamaño de las matrices, donde el tamaño de cada lado de una matriz está definido como $n = 8, 16, 32, 64, 128, 256, 512, 1024$. Los tiempos del programa secuencial se realizaron con un programa diferente al que hay de hilos y procesos, puesto que se quiere evitar contar tiempos por ejemplo la creación de procesos o de hilos y su sincronización. Los tiempos obtenidos por el problema secuencial fueron los siguientes

Cuadro I
TIEMPO DE EJECUCIÓN DEL PROGRAMA SECUENCIAL

Tamaño de la matriz	Tiempo de respuesta
8	0.001
16	0.001
32	0.001
64	0.0024
128	0.0128
256	0.1065
512	0.9322
1024	16.6686

III-A. Utilizando el método de hilos

Para la medición de las pruebas en el programa realizado con POSIX tenemos un límite de hilos que es posible lanzar, donde $h = 1, 2, 4, 8, 16$. Por lo cual para los tamaños de matriz N se resuelve la multiplicación con cada uno de los valores de h , por lo cual habrán $N \times h$ casos. El dato obtenido es el promedio de 10 pruebas realizadas para cada caso.

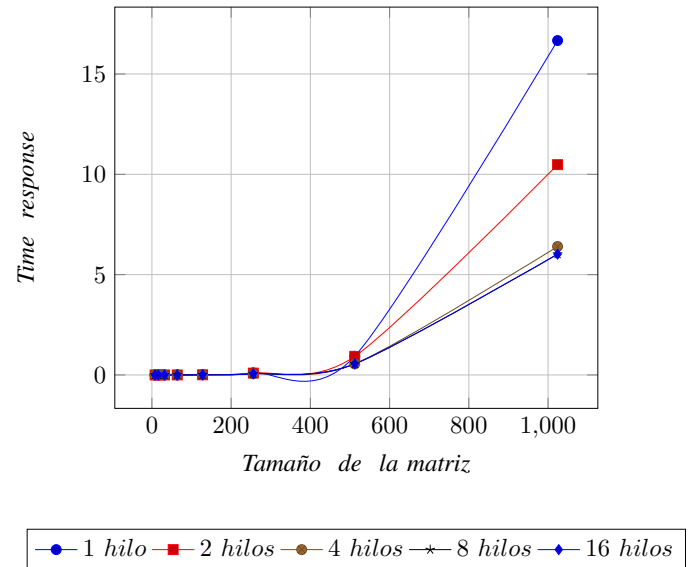
Primero vamos a observar el runtime de las distintas ejecuciones para el programa con hilos, como lo representamos en el cuadro II, donde tenemos en las filas, el tamaño de lado de las matrices que se procesan, mientras que las columnas representan la cantidad de hilos lanzados en la ejecución

Cuadro II
TIEMPO DE EJECUCIÓN CON HILOS

Tam. Matriz	1 Hilo	2 Hilos	4 Hilos	8 Hilos	16 Hilos
8	0.001	0.001	0.001	0.001	0.001
16	0.001	0.001	0.001	0.001	0.001
32	0.001	0.001	0.001	0.001	0.001
64	0.0024	0.0018	0.0017	0.0018	0.0018
128	0.0128	0.0079	0.0073	0.0073	0.0075
256	0.1065	0.0922	0.0582	0.0579	0.0579
512	0.9322	0.9264	0.5496	0.5488	0.5543
1024	16.6686	10.4812	6.4016	6.0216	6.0216

y a continuación vamos a observar el comportamiento de estos datos en una gráfica de dispersión.

Response time de hilos en función de tamaño de la matriz



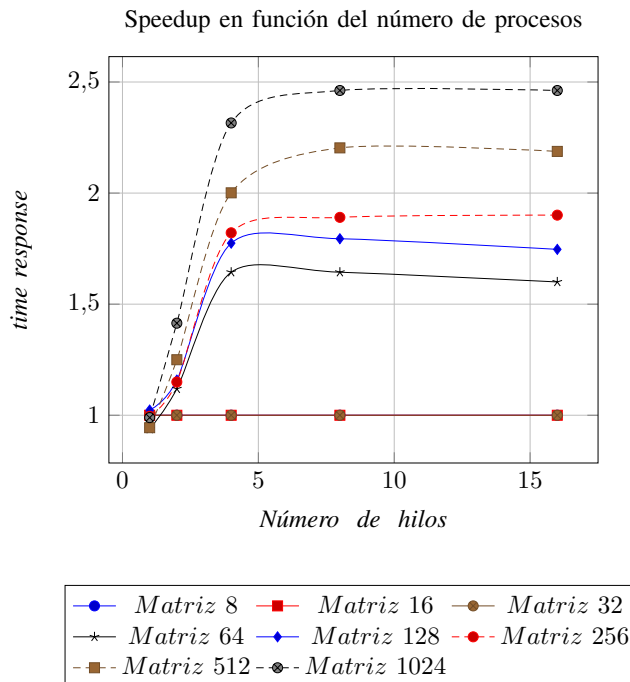
Para el cálculo del speedup o la representación del cambio en la velocidad de ejecución del programa cuando aplicamos la paralelización, usamos el mejor tiempo de ejecución del programa secuencial y se dividió entre el tiempo de ejecución obtenido con h hilos o procesos. Los resultados obtenidos fueron los siguientes:

Cuadro III
SPEEDUP DEL PROGRAMA CON HILOS

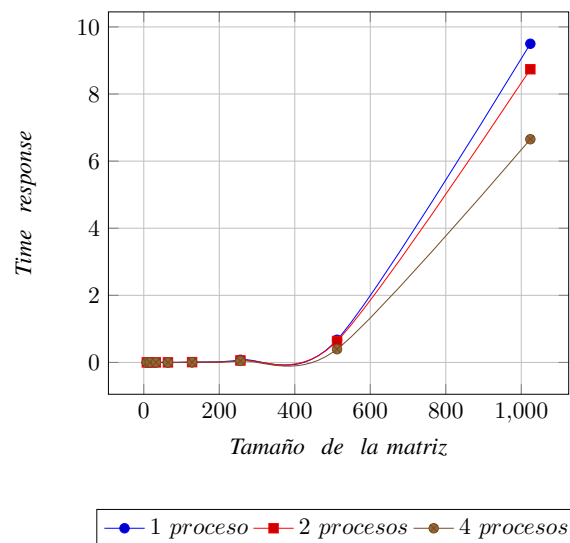
Tam. Matriz	1 hilo	2 hilos	4 hilos	8 hilos	16 hilos
8	1	1	1	1	1
16	1	1	1	1	1
32	1	1	1	1	1
64	0.937	1.119	1.644	1.644	1.645
128	1.023	1.158	1.774	1.794	1.747
256	0.995	1.150	1.821	1.891	1.901
512	0.984	1.250	2.001	2.203	2.188
1024	0.980	1.414	2.316	2.462	2.462

Estos datos los representamos gráficamente tomando el speedup calculado anteriormente en función del tamaño de la matriz, graficando los diferentes casos de número de hilos

obtenemos:



Response time de procesos en función de tamaño de la matriz



Al igual que con hilos el cálculo del speedup se calculo usando el mejor tiempo de ejecución del programa secuencial y se dividio entre el tiempo de ejecución obtenido con p procesos. Los resultados obtenidos fueron los siguientes: La

III-B. Utilizando el método de procesos

Ahora para la medición de procesos se realizaron pruebas para un número de procesos $p = 1, 2$ y 4 . Al igual que en el caso de hilos para los tamaños de matriz N se resuelve la multiplicacion con cada uno de los valores de p , por lo cual habrán Nxp casos. El dato obtenido es el promedio de 10 pruebas realizadas para cada caso.

Los tiempos de respuesta de las distintas ejecuciones del programa con procesos son presentados en el cuadro IV, dónde las filas representan el tamaño N de las matrices $N \times N$ que se procesan, mientras que las columnas representan la cantidad de procesos lanzados en la ejecución

Cuadro IV
TIEMPO DE EJECUCIÓN CON PROCESOS

Tam. Matriz	1 Proceso	2 Procesos	4 Procesos
8	0.001	0.001	0.001
16	0.001	0.001	0.001
32	0.001	0.001	0.001
64	0.002	0.0012	0.0012
128	0.009	0.0086	0.006
256	0.0772	0.0584	0.0458
512	0.675	0.6401	0.394
1024	9.498	8.738	6.652

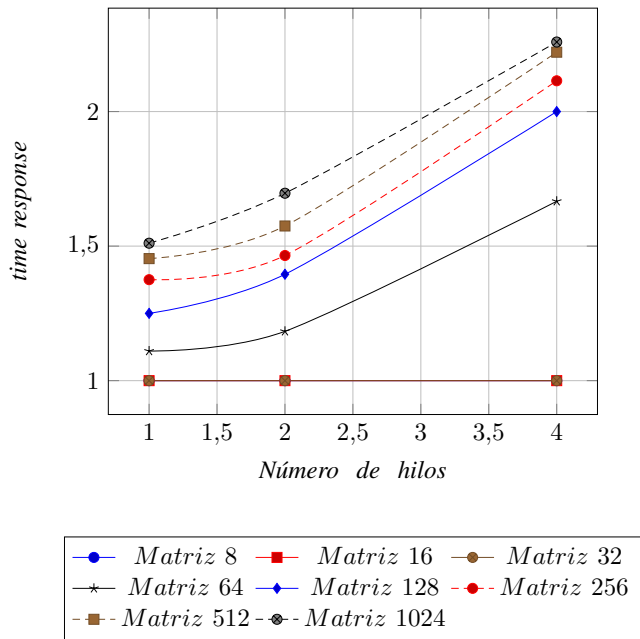
A continuación vamos a observar el comportamiento de la ejecución de los diferentes casos en la siguiente gráfica:

Cuadro V
SPEEDUP DE EJECUCIÓN CON PROCESOS

Tam. Matriz	1 Proceso	2 Procesos	4 Procesos
8	1	1	1
16	1	1	1
32	1	1	1
64	1.112	1.395	2
128	1.375	1.465	2.114
256	1.453	1.575	2.220
512	1.453	1.574	2.220
1024	1.511	1.696	2.258

representación de los datos obtenidos se grafican tomando el speedup en función del tamaño de la matriz, graficando los diferentes casos de número de procesos obtenemos:

Speedup en función del número de procesos



IV. CONCLUSIONES

Cómo podemos notar en las gráficas de hilos y procesos con respecto al tiempo de respuesta, la paralelización del problema de multiplicación de dos matrices si ofrece una mejora en la velocidad de su ejecución, un ejemplo es observar que con matrices de lado cercanos a 1000, la respuesta de ejecución es mucho menor mientras hayan más hilos que realicen procesamiento, lo mismo sucede en el caso del uso de múltiples procesos.

Aunque si observamos las gráficas hilos y procesos que representan el speedup, se presenta un límite en el cual, luego de pasarlo, la cantidad de hilos usada no generan un crecimiento en el speedup, en cambio, empieza a estabilizarse en un valor constante, lo que significaría que, después de ese punto, el despliegue de más hilos o procesos no generan una diferencia en la ejecución del programa.