

## PROGRAMACION DE MULTIPLICACIÓN DE MATRICES POR MEDIO DE HILOS

---

México Ciudad de México 18 DE MARZO DE 2022

Juan Carlos López Núñez

**Palabras Clave:**

- Hilos.
- Procesador.
- Memoria compartida.
- Multiplicación.
- Matrices.

### RESUMEN

Conforme los problemas de computo crecen el tamaño de las entradas o bien conforme la solución de los problemas requiere de un mayor numero de ciclo anidados llevando así a una mayor complejidad algorítmica surge cada vez mas la necesidad de usar técnicas de computo de alto rendimiento para poder obtener resultados en un menor tiempo posible.

### INTRODUCCION

La necesidad de procesar entradas de mayor tamaño surge conforme las bases de datos continúan creciendo o bien el tamaño del problema aumenta en su entrada. Para este proyecto se realizará una serie de pruebas para la multiplicación de matrices usando hilos por medio del modelo de memoria compartida en el lenguaje Python.

La ejecución de los hilos son dependientes del sistema y su ejecución se lleva a cabo por medio del sistema operativo.

### TECNICAS Y HERRAMIENTAS UTILIZADAS

En el siguiente programa se puede observar que la multiplicación de matrices de forma serial demanda una carga de trabajo considerable al equipo. Para las pruebas se utilizó una computadora sobre un procesador Intel Core i7 con 8 núcleos capaz de manejar 16 hilos en el sistema operativo Windows 10 de 64 bits. Las características del procesador es posible consultarlas en el caso de Windows 10 en la sección de administrador de tareas y monitor de recursos, para el caso de Linux Ubuntu 20.04 ingresar en la terminal de comandos **lscpu** para ver las características disponibles del equipo de cómputo.

Los tiempos registrados con matrices de 500 y de 1000 registraron los siguientes tiempos de procesamiento. Para desarrollar estas pruebas se utilizó el IDE Spyder 5.1.5 para el programa de multiplicación de matrices de forma serial.

```
19384, 20104, 20137, 19483, 20073, 20333, 19273, 19744, 20470, 19003, 19023, 20333, 20032, 19021, 19070,
19289, 20167, 20502, 19928, 20799, 18893, 20641, 19866, 20980, 20014, 20728, 20298, 19724, 19477, 20038,
20227, 19648, 19817, 19863, 20673, 19500, 19838, 20612, 20125, 20234, 19563, 20286, 19315, 20484, 19615,
19534, 19546, 20159, 20564, 20694, 20241, 19429, 20227, 21122, 19807, 19762, 20595, 20215, 20072, 20640,
19795, 20716, 19653, 19887, 20896, 19728, 20233, 20284, 19565, 20020]
Tiempo de ejecucion : 633.7769954204559
```

FIG1 Tiempos de ejecución en segundos en equipo con Core i7 sobre Windows para matrices de 1000X1000 con el programa de multiplicación de matrices serial.

```
10505, 10682, 10445, 10199, 10419, 10210, 10082, 9979, 10525, 10
10607, 10668, 10234, 10053, 9437, 9913, 9880, 10025, 10453, 9786
Tiempo de ejecucion : 76.26035976409912 segundos
In [13]:
```

FIG2 Tiempos de ejecución en segundos en equipo con Core i7 sobre Windows para matrices de 500X500 con el programa de multiplicación de matrices serial.

Debido a que actualmente los procesadores disponen de una arquitectura paralela es posible utilizar memoria compartida por medio de hilos para poder realizar un programa de multiplicación de matrices por medio de hilos donde cada una de las operaciones de multiplicación renglón columna se reparten entre los hilos correspondientes repartiendo de esta forma los procesos de la multiplicación de matrices para posteriormente unir los resultados correspondientes. Se utilizó el lenguaje de programación Python con la biblioteca threading.

```
MULTIPLICACION DE MATRICES PARALELAS CUADRADAS NXN

INGRESAR EL NUMERO N PARA UNA MATRIZ DE NXN : 500

INGRESAR EL NUMERO DE HILOS : 16|

Terminal de IPython
```

RESULTADOS DEL PROGRAMA DE MULTIPLICACION PARALELO

```
Tiempo de ejecucion : 143.74761295318604 segundos
```

```
In [19]:
```

```
Terminal de IPython
```

```
Historial
```

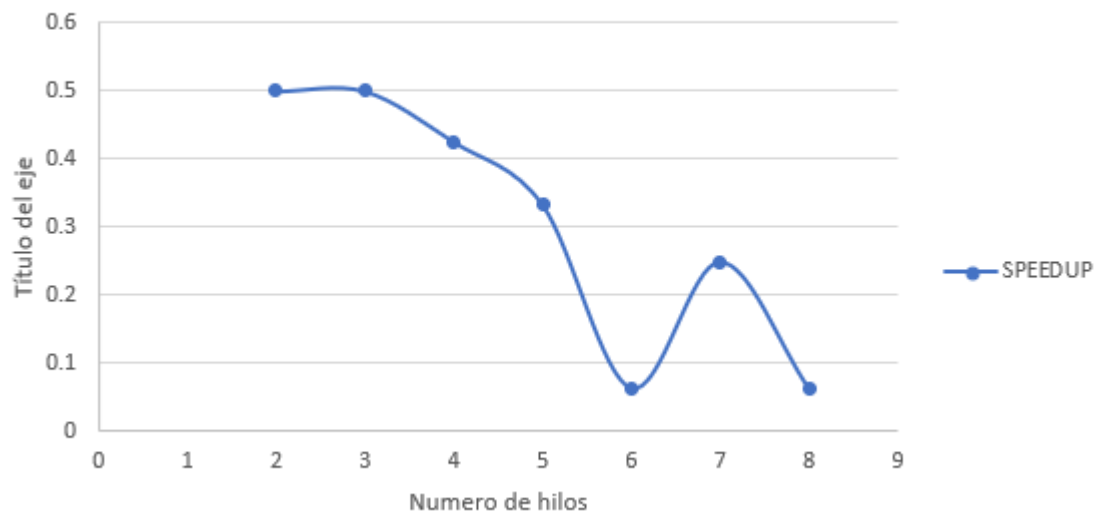
## TIEMPO DE EJECUCION DE MULTIPLICACIÓN DE MATRICES PARALELAS

### ANALISIS DE EFICIENCIA PARA MATRICES PARALELAS

En la siguiente tabla se muestran los registros de las primeras pruebas con matrices de 3X3 tomando como referencia el programa ejecutado con un solo hilo tomándolo como la referencia serial haciendo la prueba hasta con 8 hilos tomando el tiempo medido con un numero de hilos conforme se hizo la prueba. El análisis de eficiencia se hizo por medio de la formula  $\text{Eficiencia} = \frac{\text{tiempoSerial}}{(\text{tiempoParalelo} * \text{numProcesos})}$

ANALISIS DE EFICIENCIA MATRICES			
HILOS	TIEMPO PARALELO	SPEEDUP	EFICIENCIA
1	0.000997305		
2	0.001993656	0.500239177	0.250119589
3	0.001994133	0.50011956	0.16670652
4	0.002342939	0.425663987	0.106415997
5	0.002991199	0.33341304	0.066682608
6	0.015617371	0.063858696	0.010643116
7	0.003997564	0.249478142	0.035639735
8	0.015620947	0.063844076	0.007980509

SPEEDUP VS NUMERO DE HILOS



## **Análisis de complejidad**

Debido a que la función para la multiplicación de matrices tiene un ciclo anidado de 3 for la función tiene una complejidad algorítmica  $O(n^3)$  sin manejar más de un hilo. Para el caso del programa paralelo la complejidad serial se divide entre el número de hilos sobre los cuales se ejecuta el programa.  $O(n^3) / n$  donde  $n$  es el número de hilos.

## **CONCLUSIONES**

Por medio de la técnica de memoria compartida a través de hilos donde por medio de los hilos se realiza una parte de la multiplicación de matrices es posible obtener un mejor rendimiento y tiempos de respuesta menores aprovechando los recursos del procesador disponibles. Sin embargo, el reto de la programación paralela requiere de la comunicación entre los procesos para el aprovechamiento de esta técnica afectando así el tiempo de respuesta, así como hacer una repartición adecuada de las tareas de la multiplicación de la matriz.

## **REFERENCIAS**

REFERENCIAS DE MULTIPLICACION DE MATRICES CONSULTADAS EL 17 DE MARZO DE 2022

[https://rosettacode.org/wiki/Matrix\\_multiplication#C](https://rosettacode.org/wiki/Matrix_multiplication#C)

<https://codeforwin.org/2015/07/c-program-to-multiply-two-matrices.html>

<https://codescracker.com/c/program/c-program-multiply-two-matrices.htm>

<https://www.programiz.com/python-programming/examples/multiply-matrix>

<https://researchdatapod.com/python-matrix-multiplication/>

<https://www.geeksforgeeks.org/multiplication-of-matrix-using-threads/>

<https://es.stackoverflow.com/questions/166436/multiplicar-matrices-en-python>

<https://es.stackoverflow.com/questions/216439/multiplicar-matrices-python>

<https://www.geeksforgeeks.org/multiplication-two-matrices-single-line-using-numpy-python/?ref=leftbar-rightbar>

<https://www.geeksforgeeks.org/multiplication-two-matrices-single-line-using-numpy-python/?ref=leftbar-rightbar>

REFERENCIAS DE THREADING CONSULTADAS EL 17 DE MARZO DE 2022

<https://pythongeeks.org/multithreading-in-python/>

[https://www.tutorialspoint.com/python/python\\_multithreading.htm](https://www.tutorialspoint.com/python/python_multithreading.htm)

<https://www.simplifiedpython.net/python-threading-example/>

<https://realpython.com/intro-to-python-threading/>

<https://learntutorials.net/es/pthreads/topic/5669/empezando-con-pthreads>