

Informe



Presentado a:

Ramiro Andrés Barrios Valencia

Presentado por:

Santiago Valencia Díaz

Asignatura:

High Performance Computing

Universidad Tecnológica de Pereira

2024-1

Tabla de Contenido

● Resumen	3
● Introducción	3
● Marco Conceptual	3
○ High Performance Computing	3
○ Multiplicación Matricial	3
○ Complejidad Computacional	3
○ Programación Paralela	4
○ Hilos	4
○ Speedup	4
● Marco Contextual	4
○ Características de la Máquina	4
○ Desarrollo	5
○ Pruebas	5
○ Resultados de la Ejecución Secuencial	5
○ Resultados de Ejecutar el Algoritmo con Hilos	6
▪ Resultados con 2 Hilos	6
▪ Resultados con 4 Hilos	6
▪ Resultados con 8 Hilos	7
▪ Resultados con 16 Hilos	7
▪ Resultados con 32 Hilos	8
○ Resultados de la Ejecución por Procesos	8
○ Gráfico 1.	9
○ Gráfico 2.	10
● Conclusiones	11

Resumen

Este informe aborda la multiplicación de matrices como un problema de alto rendimiento, analizando su complejidad computacional y explorando la programación paralela utilizando hilos. El objetivo es optimizar el rendimiento del algoritmo de multiplicación matricial mediante la ejecución paralela en una máquina específica. Se realizan pruebas con diferentes cantidades de hilos y se evalúa el speedup obtenido.

Introducción

La multiplicación de matrices es un problema fundamental en matemáticas y computación. A medida que los conjuntos de datos crecen en tamaño, la ejecución eficiente de esta operación se vuelve esencial. En este informe, se aborda el problema de la multiplicación de matrices desde una perspectiva de High Performance Computing (HPC), utilizando la programación paralela para mejorar el rendimiento en una máquina específica.

Marco Conceptual

High Performance Computing

El High Performance Computing se refiere al uso de recursos informáticos avanzados para resolver problemas complejos y demandantes en términos de tiempo de ejecución y capacidad de procesamiento. En este informe, se explora cómo aplicar principios de HPC a la multiplicación de matrices.

Multiplicación Matricial

La multiplicación de matrices es una operación que consiste en combinar los elementos de dos matrices para formar una nueva matriz. La complejidad computacional de este proceso es $O(N^3)$, lo que significa que su tiempo de ejecución aumenta significativamente con el tamaño de las matrices.

Complejidad Computacional

La complejidad $O(N^3)$ de la multiplicación matricial impulsa la necesidad de optimizar algoritmos para matrices grandes. La programación paralela es una técnica que divide tareas en subprocesos que se ejecutan simultáneamente para mejorar el rendimiento.

Programación Paralela

La programación paralela implica dividir una tarea en partes más pequeñas y ejecutarlas simultáneamente en múltiples núcleos o procesadores. En este informe, se aplica la programación paralela utilizando hilos para realizar la multiplicación matricial.

Hilos

Los hilos son unidades de ejecución más pequeñas dentro de un proceso. Se pueden ejecutar en paralelo y comparten recursos dentro del mismo proceso. Esto permite una ejecución más eficiente de tareas en sistemas multiprocesador.

Speedup

El speedup es una medida de cuánto mejora el rendimiento al utilizar paralelismo en comparación con la ejecución secuencial. Se calcula como el tiempo de ejecución secuencial dividido por el tiempo de ejecución paralela.

Marco Contextual

Características de la Máquina

Las características de la máquina utilizada para las pruebas influyen en el rendimiento obtenido con la programación paralela. La cantidad de núcleos, la velocidad de reloj y otros factores impactan en los resultados.

Se probó el algoritmo para la multiplicación de matrices generando matrices cuadradas de dimensiones 400, 800, 1600, 3200 y 4000 respectivamente. Uno de los algoritmos fue ejecutado de manera secuencial y el otro utilizando hilos. Se analizaron los distintos métodos evaluando los recursos consumidos como el tiempo; se sacaron algunas gráficas para

mostrar los resultados obtenidos.

Características del PC donde se realizaron las pruebas:

- Procesador: Intel(R) Core(TM) i9-10900
- Cantidad de núcleos: 20
- Cantidad de subprocesos/hilos: 16
- Frecuencia básica del procesador: 2.80GHz
- Frecuencia Turbo máxima: 4.80GHz
- Ram 7.6GB DDR4 @ 2400 MHz
- SSD: 240GB - R: 540 MB/s - W: 500 MB/s
- Sistema operativo: Ubuntu 20.04.3 LTS

Desarrollo

El código implementado realiza la multiplicación de matrices utilizando la programación paralela con hilos. La cantidad de hilos utilizados se ajusta según los parámetros de entrada. Se utilizan matrices aleatorias para las pruebas.

Pruebas

Resultados de la Ejecución Secuencial

Se ejecuta el algoritmo en modo secuencial y se mide el tiempo de CPU utilizado. Esto proporciona una base para comparar con los resultados paralelos.

Tabla 1: Resultados de la ejecución secuencial

Dimensiones	SECUENCIAL				
	400	800	1600	3200	4000
1	0,21896	1,935297	21,101804	273,464547	539,594843
2	0,221405	1,926711	21,670502	267,679666	544,411861
3	0,220018	1,916417	21,284529	260,726112	540,499480

4	0,219993	1,936827	21,186428	264,411391	537,487660
5	0,219554	1,923626	21,261502	263,610806	539,650267
6	0,219276	1,945628	21,388101	257,510921	542,061808
7	0,219769	1,940097	21,365844	255,246534	544,109411
8	0,219272	1,923033	21,195518	261,046276	530,778101
9	0,219155	1,932617	21,41464	258,349886	542,711289
10	0,219456	1,940142	21,238682	262,06251	529,709401
Promedio	0,2196858	1,9320395	21,310755	262,410865	539,101412

Fuente: Elaboración Propia

Resultados de Ejecutar el Algoritmo con Hilos

Se realizan pruebas utilizando diferentes cantidades de hilos (2, 4, 8, 16 y 32). Se mide el tiempo de CPU para cada configuración y se calcula el speedup obtenido en comparación con la ejecución secuencial.

Tabla 2: Resultados de la ejecución 2 hilos

2 HILOS					
Dimensiones	400	800	1600	3200	4000
1	0,077	0,662	7,402	96,769	197,9244765
2	0,08	0,669	7,696	96,61	203,174699
3	0,08	0,678	8,148	97,682	201,373666
4	0,079	0,66	7,722	98,437	205,751219
5	0,08	0,67	7,72	97,162	212,6071965
6	0,08	0,699	7,663	101,784	197,594356
7	0,081	0,684	7,912	97,865	200,730623
8	0,079	0,65	7,569	99,446	202,7237315
9	0,079	0,666	7,455	97,424	216,570897
10	0,079	0,661	7,664	97,973	209,822679
Promedio	0,0794	0,6699	7,6951	98,1152	204,8273544
SpeedUp	2,766823678	2,884071503	2,76939286	2,67451796	1,315989785

Fuente: Elaboración Propia

Tabla 3: Resultados de la ejecución 4 hilos

4 HILOS					
Dimensiones	400	800	1600	3200	4000
1	0,042	0,337	3,885	49,083	105,7661015

2	0,045	0,353	4,274	49,571	108,7770765
3	0,045	0,352	3,88	50,473	108,2388823
4	0,046	0,353	3,937	51,329	102,5110623
5	0,044	0,356	4,036	50,627	108,4802463
6	0,046	0,355	4,03	50,515	107,777442
7	0,044	0,572	4,073	50,44	106,2407475
8	0,046	0,357	4,001	50,439	105,4981105
9	0,045	0,356	4,412	49,994	113,7955905
10	0,045	0,361	4,002	50,31	106,4887298
Promedio	0,0448	0,3752	4,053	50,2781	107,3573989
SpeedUp	4,903700893	5,149359009	5,25801999	5,21918817	1,255389516

Fuente: Elaboración Propia

Tabla 4: Resultados de la ejecución 8 hilos

8 HILOS					
Dimensiones	400	800	1600	3200	4000
1	0,025	0,179	2,026	26,772	62,927677
2	0,026	0,197	2,237	26,831	63,3025545
3	0,026	0,186	2,017	26,825	63,24969163
4	0,026	0,191	2,187	27,108	63,14125
5	0,025	0,202	2,265	27,396	62,8888085
6	0,025	0,2	2,256	27,293	62,64195675
7	0,027	0,204	2,259	27,519	63,20446613
8	0,025	0,202	2,266	28	59,76460813
9	0,026	0,205	2,278	27,819	60,18455563
10	0,027	0,21	2,292	28,096	55,727898
Promedio	0,0258	0,1976	2,2083	27,3659	61,70334663
SpeedUp	8,514953488	9,777527834	9,65029887	9,58897259	1,092123526

Fuente: Elaboración Propia

Tabla 5: Resultados de la ejecución 16 hilos

16 HILOS					
Dimensiones	400	800	1600	3200	4000
1	0,025	0,207	1,947	22,268	47,57322692
2	0,026	0,209	1,962	22,496	47,98444131
3	0,025	0,22	2,147	23,858	48,91042285
4	0,026	0,23	2,125	23,997	50,11834
5	0,026	0,232	2,158	24,334	49,99303415

6	0,028	0,236	2,179	24,211	50,93888754
7	0,025	0,236	2,156	24,189	50,88714377
8	0,025	0,233	2,229	24,282	50,90392492
9	0,028	0,277	2,298	22,781	51,33803377
10	0,026	0,234	2,164	24,055	50,89335546
Promedio	0,026	0,2314	2,1365	23,6471	49,95408107
SpeedUp	8,449453846	8,349349611	9,97461034	11,0969576	8,301491791

Fuente: Elaboración Propia

Tabla 6: Resultados de la ejecución 32 hilos

32 HILOS					
Dimensiones	400	800	1600	3200	4000
1	0,026	0,186	1,858	21,106	45,2932
2	0,025	0,191	1,953	21,156	44,762
3	0,025	0,191	1,866	21,061	43,459
4	0,026	0,193	2,045	21,852	45,1017
5	0,026	0,202	2,049	22,239	43,86262
6	0,026	0,211	2,097	22,69	43,28087
7	0,025	0,197	2,125	22,51	43,67012
8	0,026	0,212	2,129	22,442	43,578217
9	0,024	0,213	2,102	22,646	43,578217
10	0,026	0,211	2,11	22,605	44,61426
Promedio	0,0255	0,2007	2,0334	22,0307	44,1200204
SpeedUp	8,615129412	9,626504733	10,4803556	11,9111451	12,21897468

Fuente: Elaboración Propia

Resultados de ejecutar el algoritmo con procesos:

Tabla 7: Resultados de la ejecución por procesos.

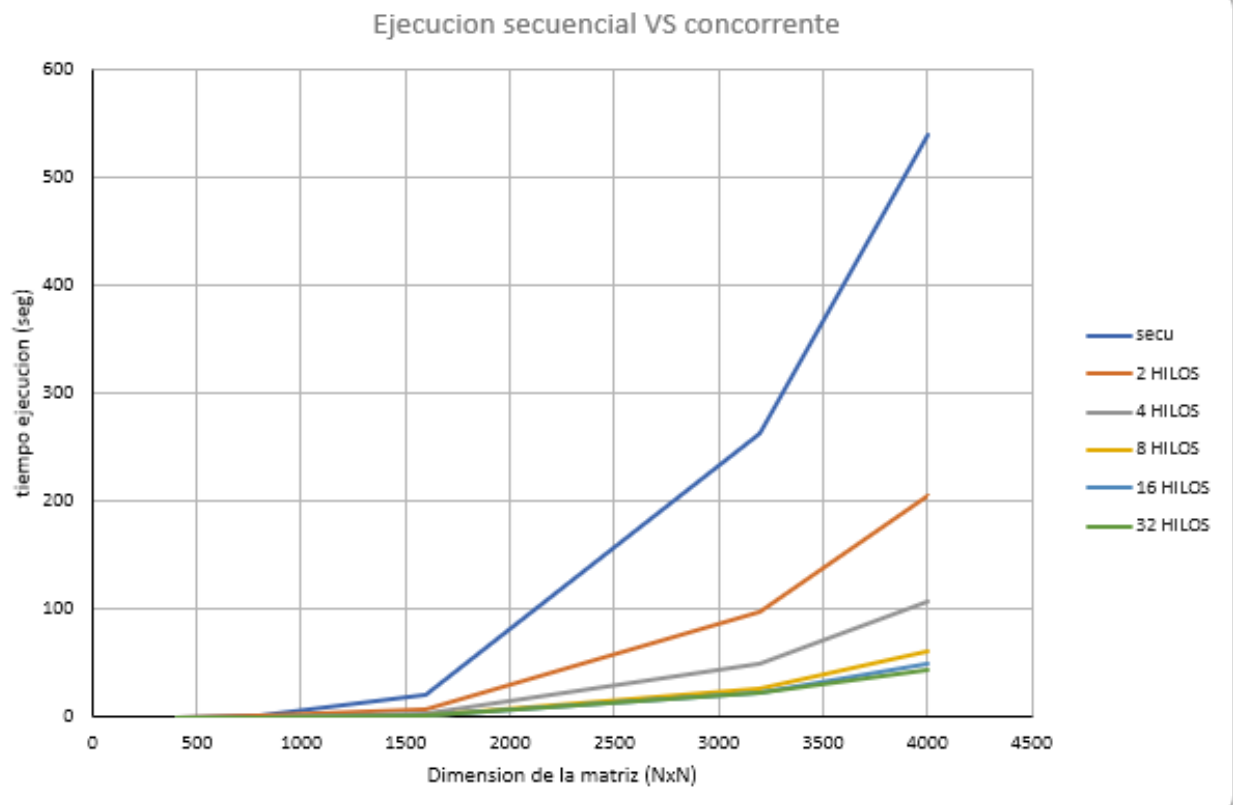
PROCESOS					
Dimensiones	400	800	1600	3200	4000
1	0,075	0,614	8,854	78,959	157,398
2	0,075	0,693	9,041	76,738	162,086
3	0,075	0,659	9,32	81,634	163,315
4	0,075	0,669	9,039	76,867	156,068
5	0,079	0,746	9,397	82,681	179,17
6	0,078	0,723	9,291	84,866	170,106
7	0,081	0,739	9,351	82,697	166,972

8	0,077	0,727	9,198	81,013	155,278
9	0,069	0,639	8,234	92,639	209,196
10	0,092	0,877	11,568	103,042	192,722
Promedio	0,0776	0,7086	9,3293	84,1136	171,2311
SpeedUp	2,831002577	2,726558707	2,28428231	3,11971982	3,148384914

Fuente: Elaboración Propia

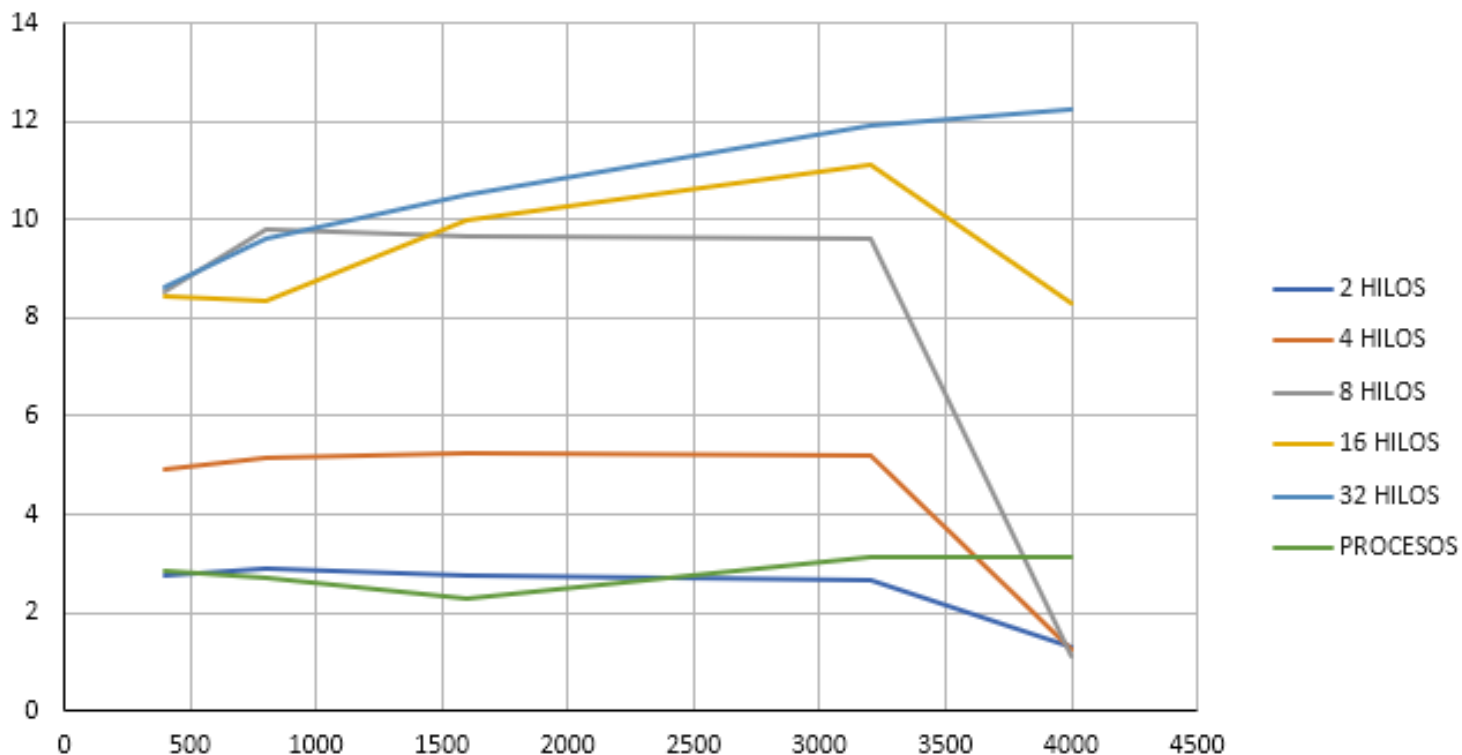
Ahora, para complementar el análisis de los datos y tener una vista un poco más amplia, se han utilizado las tablas para hacer gráficas comparativas que nos permitan visualizar la eficiencia de trabajar con hilos.

Gráfico 1. Tiempo de ejecución en función de las dimensiones de la matriz.



A continuación, se muestra una gráfica comparativa entre todos los speedup de cada método de paralelización usado, es bueno recordar que el speedup es la división del tiempo que se tiene en la ejecución secuencial, sobre el tiempo obtenido al paralelizar, por lo que en la gráfica no se incluye en speedup del método secuencial ya que este es una constante igual a 1.

Gráfico 2. Valor de Speedup en función de las dimensiones de la matriz.



Conclusiones

- La programación paralela con hilos puede mejorar significativamente el rendimiento de la multiplicación de matrices, especialmente en sistemas con múltiples núcleos. El speedup aumenta con la cantidad de hilos utilizados, pero alcanza un límite debido a la sobrecarga de administración de hilos. Se debe considerar cuidadosamente la elección de la cantidad de hilos para obtener un buen equilibrio entre rendimiento y eficiencia.
- Se puede observar que el speedup es más pronunciado a medida que aumenta el tamaño de la matriz. Esto indica que el uso de hilos o procesos es especialmente beneficioso para matrices más grandes, donde la multiplicación secuencial se vuelve prohibitivamente lenta. Por ejemplo, cabe mencionar que a partir de matrices de 1600x1600 o mayores, el uso de paralelismo (ya sea hilos o procesos) es crucial para un rendimiento óptimo.
- Se puede notar que los speedups obtenidos con hilos suelen ser mejores que con procesos en todas las dimensiones de matriz. Esto podría sugerir que la programación paralela a nivel de hilo es más eficiente en tu sistema y puede ser preferible en la mayoría de los casos.
- También es evidente que el rendimiento mejora a medida que se utilizan más hilos, pero después de cierto punto, el aumento en el speedup no es tan significativo. Por ejemplo, podría observarse que el salto de 2 hilos a 4 hilos ofrece un aumento significativo en el rendimiento, pero el aumento de 16 hilos a 32 hilos no ofrece un beneficio proporcional. Esto puede deberse a limitaciones de hardware o a la naturaleza del problema.
- El uso de hilos es esencial para matrices de gran tamaño, que la programación paralela a nivel de hilo es más eficiente que la de procesos en tu sistema, y que el número óptimo de hilos puede variar según el tamaño de la matriz y las características de la máquina.