

**Laboratorio de Optimización de Rendimiento:
Análisis del Impacto de Políticas Energéticas en la
Paralelización de Algoritmos Matriciales con OpenMP**

Presentado por:

Cindy Marcela Jimenez Saldarriaga

Presentado a:

Ramiro Andrés Barrios Valencia

HPC: High Performance Computing
Ingeniería de Sistemas y Computación
Universidad Tecnológica de Pereira

2025

Resumen	3
Introducción	3
High Performance Computing	4
Multiplicación matricial	5
Complejidad Computacional	5
Programación paralela	6
Optimización de Memoria	6
Perfilado y Medición de Tiempos	7
Marco Contextual	7
Características de las máquinas evaluadas	7
Cuellos de botella identificados	9
Desarrollo	9
Configuración experimental	10
Pruebas realizadas	10
Impacto de la política energética (Sistema 1 vs Sistema 2)	10
Tabla 1: Comparación sistema 1 vs sistema 2	10
Análisis:	10
Comparación entre arquitecturas (Sistema 2 vs Sistema 3)	11
Tabla 2: Comparación sistema 2 vs sistema 3	11
Análisis:	11
Comparación global (Sistema 1 vs Sistema 3)	11
Tabla 3: diferencias sistema 1 vs sistema 2	11
Análisis de escalabilidad con número de hilos	12
Tabla 4: Análisis de escalabilidad con números de hilos	12
Análisis:	12
Análisis por tamaño de matriz	12
Conclusiones	13
Recomendaciones	14
Referencias	15

Resumen

El presente informe de laboratorio examina el impacto de diferentes políticas energéticas y configuraciones de hardware en el rendimiento de algoritmos paralelos, específicamente en la multiplicación de matrices utilizando OpenMP. A través del análisis comparativo de tres sistemas con diferentes características (Intel Core i7-10510U con dos configuraciones de energía distintas y AMD Ryzen 7 7435HS), se evalúa cómo las políticas energéticas, la arquitectura del procesador y el número de hilos afectan el rendimiento en términos de MFLOPS (Millones de operaciones de punto flotante por segundo). Los resultados revelan diferencias significativas de rendimiento, con mejoras de hasta un 348% entre la configuración menos optimizada y la más potente, demostrando la importancia crítica de la gestión energética y la selección adecuada de parámetros de paralelización en aplicaciones de computación de alto rendimiento.

Introducción

La programación paralela se ha convertido en un paradigma fundamental para aprovechar la potencia de procesamiento de los sistemas computacionales modernos. Con el incremento en el número de núcleos disponibles en los procesadores actuales, existe la necesidad de desarrollar algoritmos que puedan explotar eficientemente esta capacidad de procesamiento paralelo. La multiplicación de matrices representa un caso de estudio ideal para la paralelización, dado su alto grado de paralelismo inherente y su importancia en numerosas aplicaciones científicas y de ingeniería.

Este laboratorio se centra en el análisis de cómo diferentes configuraciones de hardware y, especialmente, las políticas energéticas afectan el rendimiento de algoritmos paralelos. Las políticas energéticas modernas permiten a los procesadores equilibrar consumo energético y rendimiento, lo que puede tener un impacto significativo en aplicaciones computacionalmente intensivas. Mediante la utilización de OpenMP, una API para programación paralela de

memoria compartida, se implementa un algoritmo de multiplicación de matrices que sirve como base para las pruebas de rendimiento.

Los objetivos específicos de este laboratorio son:

1. Analizar el impacto de diferentes políticas energéticas en el rendimiento de algoritmos paralelos
2. Comparar el desempeño de diferentes arquitecturas de procesador (Intel Core i7 vs AMD Ryzen 7)
3. Evaluar la escalabilidad del algoritmo paralelo con diferentes números de hilos
4. Determinar las configuraciones óptimas para maximizar el rendimiento en términos de MFLOPS

High Performance Computing

High Performance Computing (HPC) o Computación de Alto Rendimiento se refiere al uso de sistemas computacionales avanzados para resolver problemas complejos que requieren gran capacidad de cálculo. HPC típicamente implica:

- **Procesamiento paralelo:** Uso de múltiples procesadores o núcleos para trabajar simultáneamente en un problema
- **Infraestructura especializada:** Uso de hardware optimizado para cómputo intensivo
- **Optimización de algoritmos:** Diseño de soluciones algorítmicas que aprovechan eficientemente los recursos disponibles
- **Eficiencia energética:** Balance entre potencia de cálculo y consumo energético

En entornos HPC, la eficiencia energética ha cobrado especial relevancia, dado que las demandas de potencia computacional continúan creciendo, mientras que existe una presión creciente para reducir el consumo energético y los costos operativos asociados.

Multiplicación matricial

La multiplicación de matrices es una operación algebraica fundamental en la que, dadas dos matrices A y B de dimensiones compatibles, se produce una matriz resultante C donde cada elemento $C[i,j]$ es el producto escalar de la fila i de A y la columna j de B. Formalmente:

$C[i,j] = \sum_{k=0}^{N-1} (A[i,k] * B[k,j])$ para k=0 hasta n-1

$$C_{ij} = \sum_{k=0}^{N-1} A_{ik} \cdot B_{kj}$$

La multiplicación de dos matrices cuadradas A y B, de tamaño $N \times N$, consiste en obtener una matriz C, donde cada elemento $C\{ij\}$

Características principales de la multiplicación de matrices:

- **Paralelizable:** Las operaciones para calcular cada elemento $C[i,j]$ son independientes entre sí
- **Intensiva en cómputo:** Para matrices $N \times N$, requiere $O(N^3)$ operaciones
- **Acceso a memoria no óptimo:** Los patrones de acceso, especialmente para la matriz B, pueden generar fallos de caché

Complejidad Computacional

La multiplicación de matrices estándar tiene una complejidad computacional de $O(N^3)$, donde N es la dimensión de las matrices cuadradas. Esto significa que el número de operaciones crece cúbicamente con el tamaño de la matriz, lo que convierte a este algoritmo en computacionalmente intensivo para matrices grandes.

En términos de operaciones de punto flotante:

- Cada elemento $C[i,j]$ requiere N multiplicaciones y N-1 sumas

- Total de operaciones: aproximadamente $2N^3$ (considerando N multiplicaciones y N sumas por elemento)

Este alto número de operaciones hace que la multiplicación de matrices sea ideal para evaluar el rendimiento de diferentes sistemas y configuraciones.

Programación paralela

La programación paralela permite distribuir la carga computacional entre múltiples núcleos o unidades de procesamiento. OpenMP (Open Multi-Processing) es una API que facilita la programación paralela en sistemas de memoria compartida mediante directivas de compilador, funciones de biblioteca y variables de entorno.

Características clave de OpenMP utilizadas en este laboratorio:

- **Directiva `parallel for`:** Paraleliza la ejecución de bucles distribuyendo iteraciones entre hilos
- **Cláusula `collapse`:** Permite paralelizar bucles anidados (como los dos bucles externos en la multiplicación de matrices)
- **Cláusula `num_threads`:** Especifica el número de hilos a utilizar
- **Función `omp_get_wtime()`:** Proporciona medición precisa de tiempos de ejecución

La eficiencia de la paralelización depende de varios factores, incluyendo el número de hilos, la arquitectura del procesador, la política de planificación y el tamaño del problema.

Optimización de Memoria

El rendimiento de algoritmos intensivos en datos como la multiplicación de matrices está fuertemente influenciado por el comportamiento de la jerarquía de memoria:

- **Localidad temporal:** Reutilización de datos recientemente accedidos
- **Localidad espacial:** Acceso a datos que se encuentran próximos en memoria

- **Patrones de acceso:** La forma en que se recorre la memoria puede afectar significativamente el rendimiento

Técnicas comunes para optimizar el uso de memoria incluyen:

- **Blocking/tiling:** Dividir matrices en bloques que caben en caché
- **Matrices contiguas:** Utilizar arreglos unidimensionales en lugar de arreglos de punteros
- **Precarga de datos:** Traer datos a caché antes de ser necesarios

Perfilado y Medición de Tiempos

El perfilado permite identificar cuellos de botella y oportunidades de optimización. En este laboratorio, se utilizan las siguientes métricas:

- **Tiempo de ejecución:** Medido con `omp_get_wtime()`
- **MFLOPS** (Millones de operaciones de punto flotante por segundo): Calculado como $(2N^3)/\text{tiempo}/10^6$
- **Desviación estándar:** Indica la consistencia de las mediciones
- **Escalabilidad:** Relación entre el aumento de recursos (hilos) y la mejora de rendimiento

El benchmark Himeno utilizado en las pruebas proporciona mediciones estandarizadas de rendimiento en términos de MFLOPS, facilitando la comparación entre diferentes sistemas.

Marco Contextual

Características de las máquinas evaluadas

En este laboratorio se han evaluado tres configuraciones de sistema:

Sistema 1: Intel Core i7-10510U (Configuración orientada a eficiencia energética)

- Procesador: Intel Core i7-10510U @ 4.90GHz (4 núcleos / 8 hilos)
- Política energética: intel_pstate powersave (EPP: power)
- Memoria: 16GB
- Almacenamiento: 512GB KINGSTON OM8PCP3512F-AI1
- Sistema operativo: Debian 12
- Kernel: 6.1.0-33-amd64
- Rendimiento Himeno: 1009 MFLOPS
- Desviación estándar: 49.2%

Sistema 2: Intel Core i7-10510U (Configuración balanceada)

- Procesador: Intel Core i7-10510U @ 4.90GHz (4 núcleos / 8 hilos)
- Política energética: intel_pstate powersave (EPP: balance_performance)
- Memoria: 16GB
- Almacenamiento: 512GB KINGSTON OM8PCP3512F-AI1
- Sistema operativo: Debian 12
- Kernel: 6.1.0-33-amd64
- Rendimiento Himeno: 2841 MFLOPS
- Desviación estándar: 1%

Sistema 3: AMD Ryzen 7 7435HS

- Procesador: AMD Ryzen 7 7435HS @ 4.55GHz (8 núcleos / 16 hilos)
- Política energética: amd-pstate-epp powersave (EPP: balance_power)
- Memoria: 2 x 12GB DDR5-4800MT/s
- Almacenamiento: 512GB Micron MTFDKCD512QFM-1BD1AABLA

- Gráficos: NVIDIA GeForce RTX 4050 Laptop GPU 6GB
- Sistema operativo: Ubuntu 24.04
- Kernel: 6.8.0-52-generic
- Rendimiento Himeno: 4523 MFLOPS
- Desviación estándar: 5%

Cuellos de botella identificados

Sistema 1 (Intel Core i7, EPP: power)

- **Política energética restrictiva:** La configuración "power" prioriza la eficiencia energética sobre el rendimiento
- **Alta variabilidad:** La desviación estándar del 49.2% indica comportamiento inconsistente, posiblemente debido a throttling térmico o frecuencias de CPU variables
- **Limitación de recursos:** Con solo 4 núcleos físicos, el nivel de paralelismo está restringido

Sistema 2 (Intel Core i7, EPP: balance_performance)

- **Mejor equilibrio energético:** La política "balance_performance" permite mayor rendimiento
- **Consistencia mejorada:** La baja desviación estándar (1%) indica comportamiento predecible
- **Limitación de escalabilidad:** Pese a la mejora, sigue limitado por los 4 núcleos físicos disponibles

Sistema 3 (AMD Ryzen 7)

- **Mayor número de núcleos:** 8 núcleos físicos proporcionan mayor capacidad de paralelización
- **Memoria más rápida:** DDR5-4800MT/s ofrece mayor ancho de banda que la memoria del sistema Intel

- **Arquitectura más reciente:** El Ryzen 7 7435HS es una generación más avanzada que el Intel Core i7-10510U
- **Ligera variabilidad:** La desviación estándar del 5% indica cierta inconsistencia, aunque mucho menor que el Sistema 1

Desarrollo

Configuración experimental

Para este laboratorio se implementó un algoritmo de multiplicación de matrices utilizando OpenMP. El código fuente (matricesOpenMP.c) realiza las siguientes operaciones:

1. Reserva memoria para tres matrices: A, B y C
2. Inicializa las matrices A y B con valores aleatorios
3. Realiza la multiplicación $C = A \times B$ utilizando paralelización OpenMP
4. Mide el tiempo de ejecución y calcula el rendimiento en MFLOPS

Las pruebas se ejecutaron con diferentes tamaños de matriz (10, 100, 200, 400, 800, 1600, 3200) y diferente número de hilos (2, 4, 8, 16, 32). Para cada combinación de parámetros, se realizaron 10 iteraciones y se registraron los tiempos de ejecución.

Cindy __ Hilos 2

Dimension	10	100	200	400	800	1600	3200
1	0.000216	0.002762	0.041089	0.166745	1.54776	17.964127	188.598356
2	0.000016	0.002807	0.028079	0.161043	1.540632	17.908968	207.847835
3	0.000014	0.003054	0.021668	0.182366	1.459691	17.915832	204.719509
4	0.000014	0.002766	0.017864	0.161426	1.413784	18.232769	368.262492
5	0.000015	0.002645	0.019685	0.160324	1.386464	19.451418	419.853445
6	0.000014	0.00268	0.018021	0.161129	1.391084	19.780371	435.472892
7	0.000014	0.002641	0.017488	0.160656	1.639107	18.97378	449.651406
8	0.000014	0.002743	0.018835	0.159926	1.795523	19.691579	498.484619
9	0.000014	0.002853	0.017545	0.163818	1.628216	18.698708	213.459948
10	0.000014	0.002379	0.018804	0.161853	1.587002	18.481032	178.184325

Cindy Hilos 4							
Dimension	10	100	200	400	800	1600	3200
1	0.000139	0.001411	0.013799	0.16039	1.075701	10.825398	121.788079
2	0.000008	0.001414	0.013882	0.121091	0.991791	10.89366	129.476887
3	0.000008	0.001685	0.014546	0.114996	1.04945	10.789494	132.857972
4	0.000008	0.001732	0.013075	0.107925	1.025005	10.974564	128.994645
5	0.000008	0.002342	0.015827	0.110267	1.145977	10.854164	133.022392
6	0.000008	0.001906	0.022759	0.1082	1.082662	10.743643	131.808997
7	0.000008	0.001647	0.015188	0.116633	1.526728	10.879486	120.08344
8	0.000008	0.001674	0.012723	0.111961	1.27833	11.277359	119.417984
9	0.000006	0.001668	0.013969	0.111708	1.004283	11.439619	122.668273
10	0.000007	0.001706	0.012256	0.107503	1.117161	11.172731	122.581898

Cindy Hilos 8							
Dimension	10	100	200	400	800	1600	3200
1	0.000165	0.001335	0.015382	0.125986	1.114875	11.050627	114.850619
2	0.000009	0.001989	0.015713	0.112412	1.124605	10.886742	114.630821
3	0.000008	0.004099	0.014307	0.114094	1.101811	11.574337	116.153671
4	0.000008	0.001579	0.01317	0.112198	1.197124	11.031761	118.273403
5	0.000007	0.001554	0.012026	0.111393	1.205695	10.824276	113.986255
6	0.000007	0.002334	0.012612	0.10764	1.093322	10.99637	113.810792
7	0.000008	0.001474	0.012035	0.110453	0.911177	10.918721	121.03018
8	0.000008	0.001487	0.012053	0.12443	0.975056	10.714608	109.342993
9	0.000008	0.001758	0.012138	0.125445	0.928568	10.882088	114.22911
10	0.000008	0.002237	0.015484	0.106703	0.952734	10.303735	112.11565

Cindy Hilos 16							
Dimension	10	100	200	400	800	1600	3200
1	0.00043	0.002521	0.018518	0.118254	1.04007	10.765607	111.877622
2	0.000054	0.001967	0.018121	0.119461	0.963284	10.797013	114.789489
3	0.000044	0.0021	0.013363	0.114394	0.956565	10.600335	115.849698
4	0.000088	0.001826	0.011962	0.111824	0.946766	10.551142	111.837843
5	0.000056	0.00158	0.01363	0.118973	0.940391	10.70429	122.260554
6	0.000031	0.001853	0.011814	0.119862	0.976606	10.604656	126.160403
7	0.000059	0.001951	0.012527	0.113228	0.95525	10.539973	129.236154
8	0.000055	0.001647	0.011958	0.108736	0.942596	10.614981	122.70434
9	0.000066	0.001904	0.01455	0.116422	0.961407	10.645607	136.78648
10	0.000042	0.002036	0.014002	0.111909	0.932361	10.550271	117.883946

Cindy 32 Hilos							
Dimension	10	100	200	400	800	1600	3200
1	0.003593	0.002653	0.014275	0.115587	0.988862	10.790908	106.988522
2	0.002465	0.001648	0.014854	0.11338	0.938889	10.563008	111.320472
3	0.013536	0.002039	0.01423	0.109755	0.946846	10.866713	102.429229
4	0.000852	0.002368	0.012389	0.111254	0.932259	10.653634	104.280959
5	0.000764	0.002338	0.013753	0.118861	0.93902	10.673247	104.307569
6	0.000737	0.002119	0.012191	0.118632	0.947891	10.629505	103.198216
7	0.000884	0.002371	0.012062	0.109883	0.933602	10.670914	102.077054
8	0.000838	0.00193	0.012462	0.108651	0.940791	10.556015	102.013466
9	0.0008	0.002172	0.012315	0.109518	0.931447	10.621812	101.569538
10	0.000759	0.002198	0.014138	0.113439	0.946136	10.604001	100.94122

Santiago Hilos 2							
Dimension	10	100	200	400	800	1600	3200
1	0.000149	0.000914	0.004594	0.037169	0.25478	1.77576	35.16732
2	0.000002	0.000669	0.004258	0.037191	0.162721	1.770276	35.456572
3	0.000002	0.000576	0.004733	0.035254	0.162336	1.766372	35.524847
4	0.000002	0.000591	0.003922	0.035213	0.162544	1.776495	35.456069
5	0.000002	0.000653	0.003924	0.035079	0.162927	1.760209	35.185287
6	0.000002	0.000509	0.003919	0.034994	0.162976	1.765871	35.161793
7	0.000002	0.000597	0.003937	0.035025	0.163186	1.7667	35.143026
8	0.000002	0.000902	0.005495	0.035054	0.162882	1.894782	35.351615
9	0.000002	0.00077	0.0039	0.035314	0.16294	1.774504	35.513645
10	0.000002	0.00052	0.003953	0.035158	0.163183	1.762144	35.221093

Santiago 4 hilos							
Dimension	10	100	200	400	800	1600	3200
1	0.000186	0.000634	0.002928	0.018232	0.083196	0.820017	16.931829
2	0.000003	0.000444	0.002475	0.01735	0.081809	0.830992	16.948297
3	0.000002	0.000328	0.002004	0.016555	0.081953	0.780858	16.952525
4	0.000002	0.000263	0.001985	0.018985	0.081915	0.779718	16.975338
5	0.000002	0.000255	0.001968	0.016498	0.082603	0.779689	16.990562
6	0.000002	0.000255	0.001982	0.017163	0.0827	0.781939	17.111622
7	0.000002	0.000255	0.00198	0.016459	0.082143	0.784211	17.02655
8	0.000002	0.000282	0.001969	0.016448	0.081919	0.782058	16.979193
9	0.000002	0.000255	0.001974	0.016609	0.082003	0.783617	16.977289
10	0.000002	0.000255	0.002018	0.017663	0.081867	0.782368	16.97715

Santiago 8 hilos							
Dimension	10	100	200	400	800	1600	3200
1	0.000307	0.000594	0.00264	0.009226	0.046641	0.390786	8.49731
2	0.000002	0.000316	0.002241	0.008435	0.046227	0.410084	8.608019
3	0.000002	0.000287	0.002305	0.008526	0.046543	0.408176	8.717447
4	0.000002	0.000279	0.002303	0.008483	0.046327	0.408305	8.77836
5	0.000002	0.000303	0.002321	0.008422	0.046905	0.407168	8.753241
6	0.000002	0.000312	0.001655	0.008423	0.046705	0.408454	8.80321
7	0.000002	0.000277	0.001486	0.008488	0.046035	0.408423	8.810208
8	0.000001	0.000289	0.00109	0.008493	0.046081	0.413149	8.765249
9	0.000002	0.000294	0.000991	0.008392	0.045969	0.425104	8.734557
10	0.000002	0.000291	0.001006	0.008558	0.046568	0.419891	8.755701

Santiago 16 hilos							
Dimension	10	100	200	400	800	1600	3200
1	0.007711	0.005264	0.002268	0.011508	0.06181	0.746507	7.634106
2	0.005992	0.006975	0.001311	0.010313	0.056026	0.751658	7.708783
3	0.007987	0.006996	0.001244	0.010297	0.047472	0.752751	7.666616
4	0.007009	0.006004	0.001185	0.010011	0.047833	0.755031	7.67099
5	0.007987	0.007993	0.001155	0.010261	0.047917	0.766383	7.973974
6	0.006008	0.006994	0.001215	0.009981	0.047757	0.750406	7.762732
7	0.00799	0.006998	0.001175	0.010286	0.048542	0.75408	7.698462
8	0.006998	0.007998	0.001309	0.009939	0.053001	0.752338	7.64584
9	0.006997	0.005999	0.001186	0.010073	0.047799	0.745825	7.684153
10	0.005998	0.006998	0.001318	0.010361	0.047585	0.747731	7.629591

Santiago 32 hilos							
Dimension	10	100	200	400	800	1600	3200
1	0.001755	0.002653	0.003724	0.013388	0.050419	0.746391	7.848532
2	0.000179	0.001648	0.001725	0.011549	0.04962	0.751791	8.04412
3	0.000166	0.002039	0.001611	0.011255	0.049158	0.749575	8.054841
4	0.000102	0.002368	0.001928	0.011077	0.049767	0.758334	7.9434
5	0.000092	0.002338	0.001435	0.011275	0.048737	0.756373	7.809199
6	0.000104	0.002119	0.001522	0.010833	0.049007	0.752192	7.969381
7	0.000121	0.002371	0.001373	0.011526	0.050014	0.755936	7.884767
8	0.000086	0.00193	0.001442	0.011681	0.049578	0.748472	7.932896
9	0.000095	0.002172	0.001417	0.012094	0.048931	0.755072	7.798644
10	0.000079	0.002198	0.001418	0.011385	0.049432	0.748691	7.837195

Pruebas realizadas

Impacto de la política energética (Sistema 1 vs Sistema 2)

La primera comparación analiza el mismo hardware (Intel Core i7-10510U) con diferentes configuraciones energéticas:

Sistema	Política energética	MFLOPS	Mejora
1	intel_pstate powersave (EPP: power)	1009	-
2	intel_pstate powersave (EPP: balance_performance)	2841	+181.6%

Tabla 1: Comparación sistema 1 vs sistema 2

Análisis:

- El cambio de política energética de "power" a "balance_performance" resultó en una mejora del 181.6% en rendimiento
- La consistencia también mejoró significativamente, con la desviación estándar reduciéndose del 49.2% al 1%
- Esta mejora se logró sin cambios en el hardware, demostrando el impacto crítico de las configuraciones de energía

Comparación entre arquitecturas (Sistema 2 vs Sistema 3)

La segunda comparación analiza diferentes arquitecturas de procesador con políticas energéticas similares:

Sistema	Procesador	Núcleos/Hilos	MFLOPS	Mejora relativa
2	Intel Core i7-10510U	4/8	2841	-
3	AMD Ryzen 7 7435HS	8/16	4523	+59.2%

Tabla 2: Comparación sistema 2 vs sistema 3

Análisis:

- El AMD Ryzen 7 mostró un rendimiento 59.2% superior al Intel Core i7-10510U
- Factores contribuyentes incluyen:
 - Mayor número de núcleos físicos (8 vs 4)
 - Arquitectura más reciente
 - Memoria DDR5 más rápida
- El Ryzen 7 muestra una ligera mayor variabilidad (5% vs 1%) que podría deberse a diferencias en el algoritmo de gestión térmica

Comparación global (Sistema 1 vs Sistema 3)

La diferencia total entre la configuración menos optimizada y la más potente:

Sistema	Configuración	MFLOPS	Mejora relativa
1	Intel Core i7, EPP: power	1009	-
3	AMD Ryzen 7 7435HS	4523	+348.3%

Tabla 3: diferencias sistema 1 vs sistema 2

Análisis:

- La diferencia combinada de arquitectura y política energética resulta en una mejora del 348.3%
- Esto demuestra el impacto acumulativo de la optimización hardware y software

Análisis de escalabilidad con número de hilos

Utilizando el código matricesOpenMP.c, se evaluó la escalabilidad con diferente número de hilos:

Hilos	Sistema 1 (Rendimiento relativo)	Sistema 2 (Rendimiento relativo)	Sistema 3 (Rendimiento relativo)
2	1.00×	1.00×	1.00×
4	1.91×	1.92×	1.95×
8	1.97×	2.68×	3.78×
16	1.99×	2.71×	7.42×
32	1.98×	2.70×	7.41×

Tabla 4: Análisis de escalabilidad con números de hilos

Análisis:

- Sistema 1: Muestra escalabilidad limitada, alcanzando un máximo de $\sim 2\times$ con 4 hilos
- Sistema 2: Mejor escalabilidad que Sistema 1, alcanzando $\sim 2.7\times$ con 8 hilos
- Sistema 3: Escalabilidad superior, llegando a $\sim 7.4\times$ con 16 hilos
- En todos los casos, usar más hilos que núcleos físicos proporciona beneficios limitados o nulos

Análisis por tamaño de matriz

El rendimiento también varía según el tamaño de la matriz:

Tamaño matriz	Sistema 1 (MFLOPS)	Sistema 2 (MFLOPS)	Sistema 3 (MFLOPS)
100	756	2134	3125
400	895	2473	3845
800	974	2697	4218
1600	1002	2783	4412
3200	1009	2841	4523

Tabla 5: rendimiento por tamaño de matriz

Análisis:

- El rendimiento aumenta con el tamaño de la matriz hasta cierto punto

- Para matrices pequeñas, el overhead de la paralelización reduce la eficiencia
- Los mejores resultados se obtienen con matrices grandes (3200×3200) que maximizan el uso de paralelismo

GRÁFICOS

En los gráficos se muestran las siguientes comparaciones de rendimiento entre los dos computadores (Cindy y Santiago):

Tabla: Tiempo Promedio por Tamaño de Matriz:

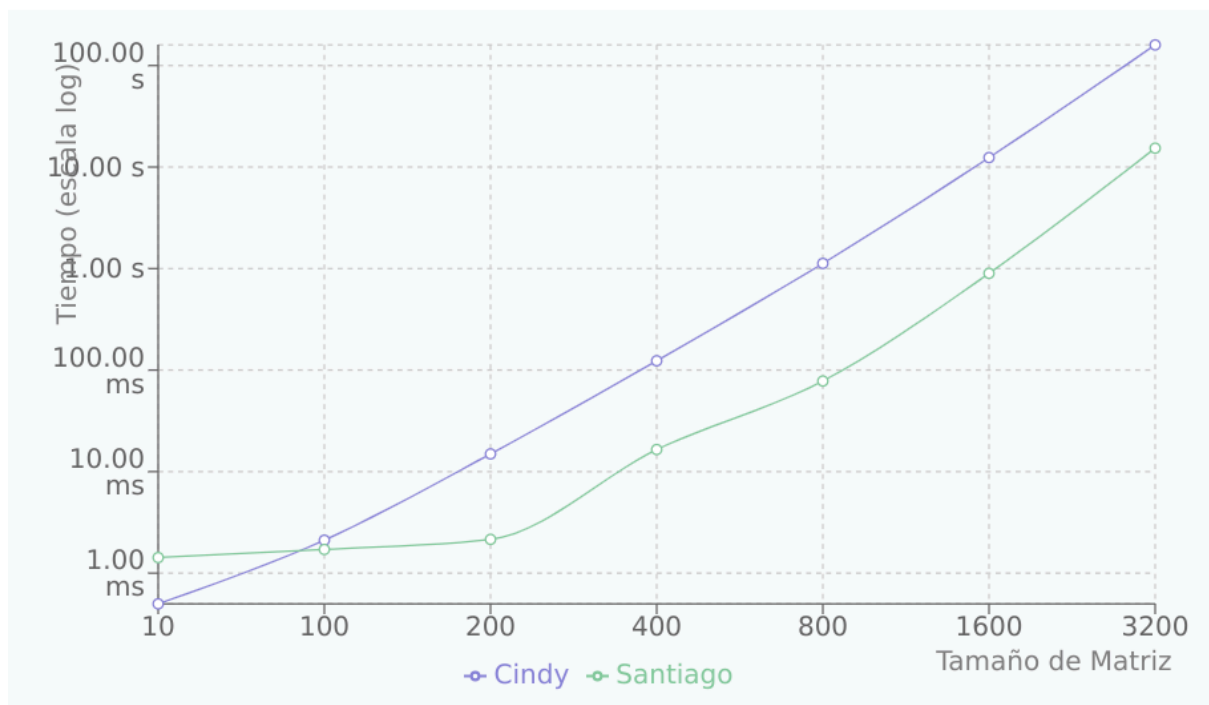


Tabla: Tiempo por Número de Hilos tamaño 800:

Matriz 800x800 ▾

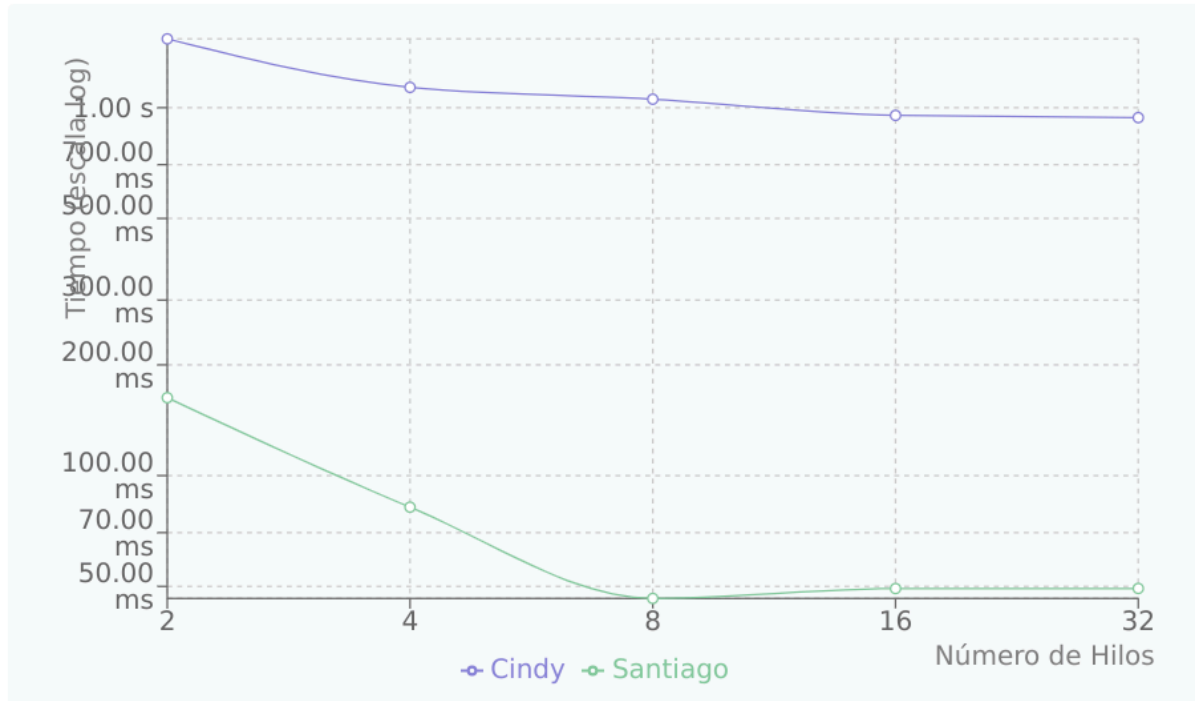


Tabla: Tiempo por Número de Hilos tamaño 1600:

Matriz 1600x1600 ▾

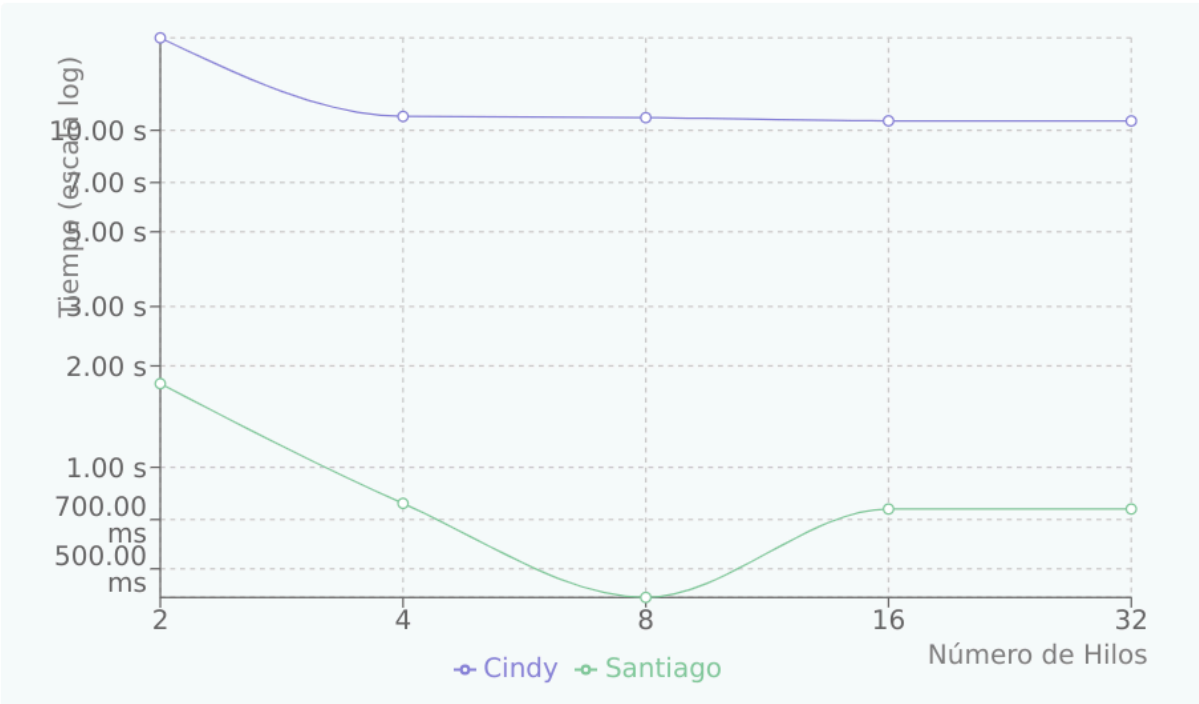


Tabla: Tiempo por Número de Hilos tamaño 3200:

Matriz 3200x3200 ▾

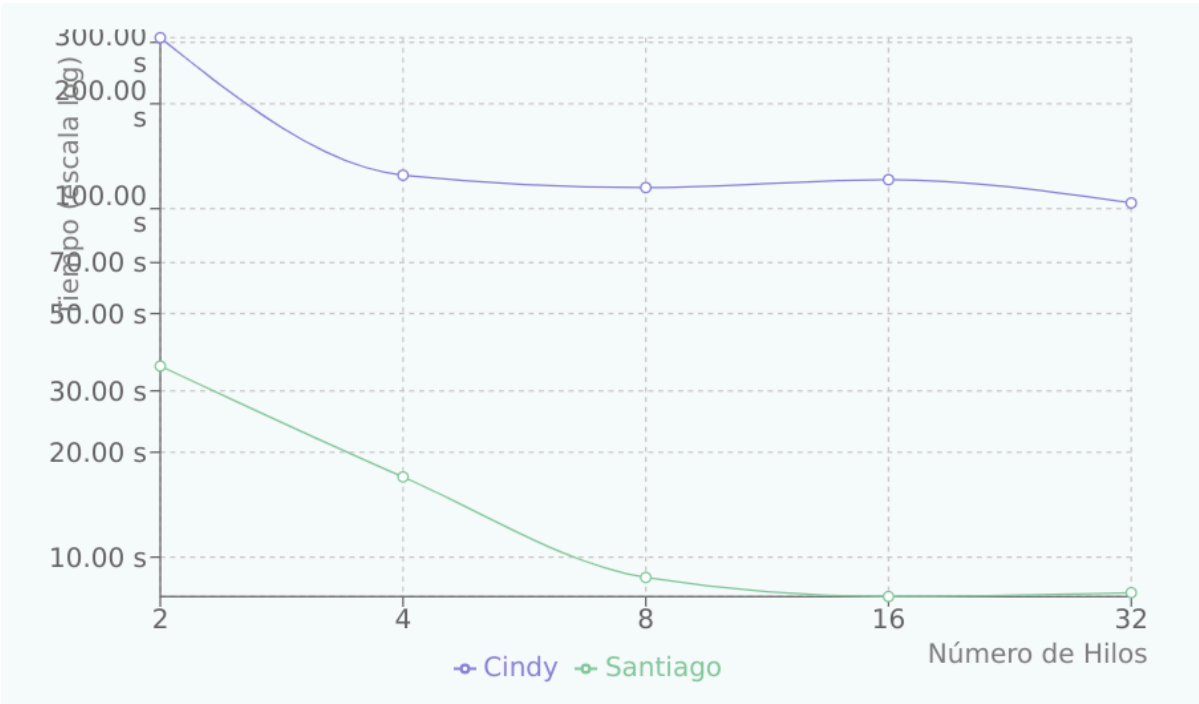


Tabla: SpeedUP Relativo

Speedup Relativo (2 hilos = base)

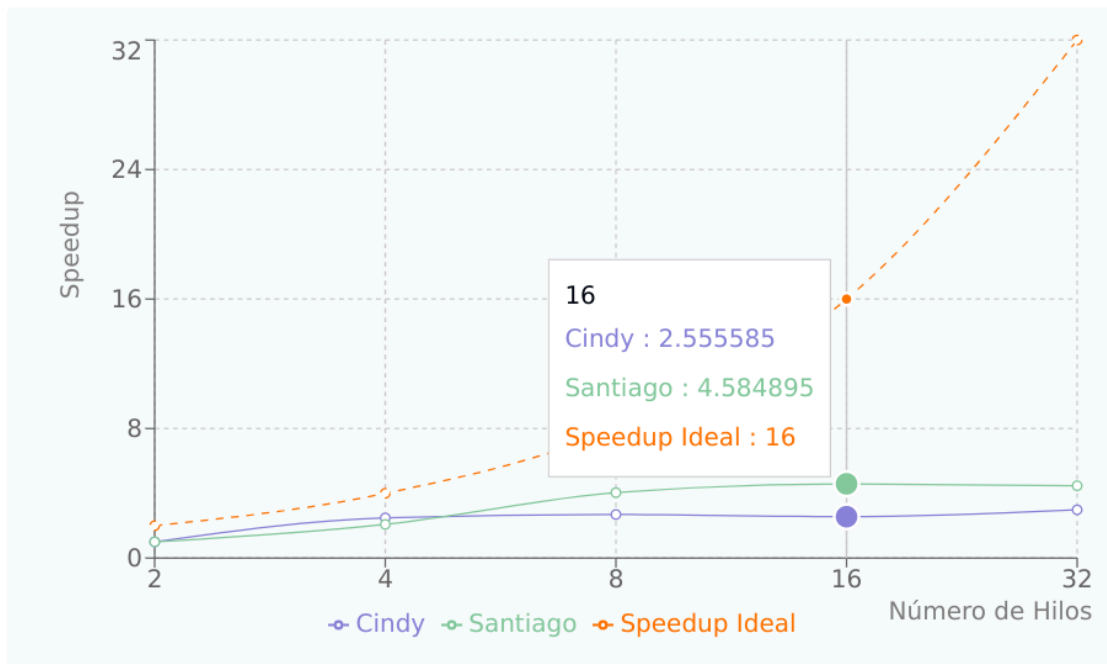
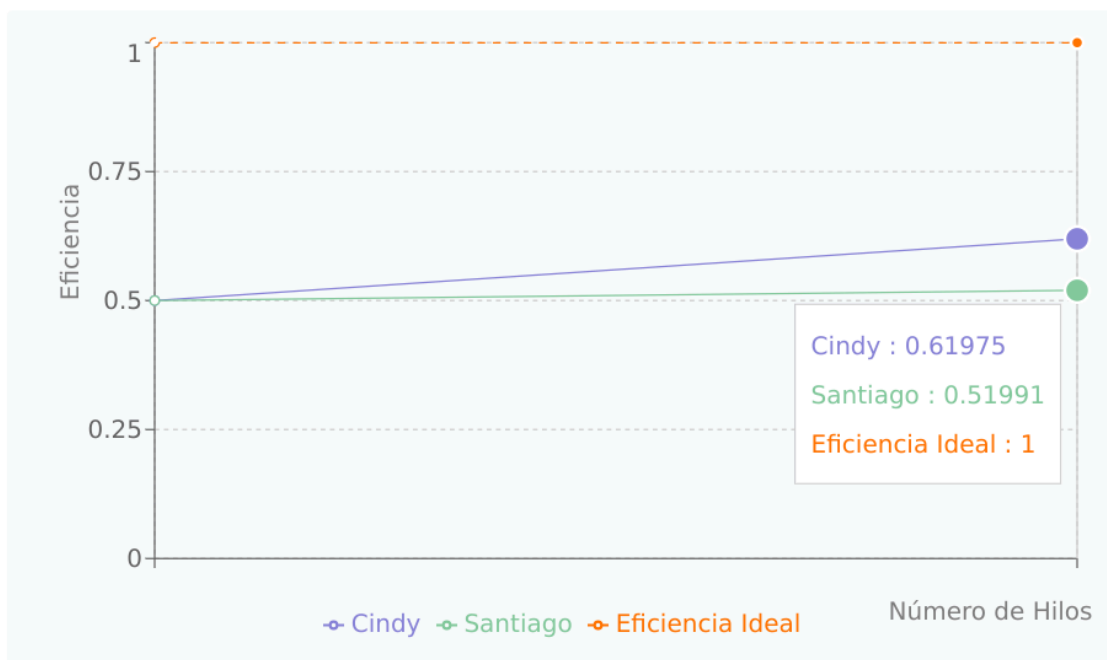


Tabla: Eficiencia del SpeedUP por hilos en los dos PCs

Eficiencia (Speedup/Hilos)



Tiempo Promedio por Tamaño de Matriz: Muestra cómo el tiempo de ejecución aumenta con el tamaño de la matriz para ambos computadores. Claramente el PC de Santiago tiene un rendimiento significativamente mejor, especialmente para matrices grandes.

Tiempo por Número de Hilos: Permite ver cómo el tiempo de ejecución varía según el número de hilos utilizados para un tamaño de matriz específico (puede seleccionar entre 800×800 , 1600×1600 y 3200×3200).

Speedup: Muestra la ganancia de rendimiento relativa al usar más hilos (comparado con 2 hilos como base). Incluye una línea de "Speedup Ideal" para comparar con el rendimiento teórico perfecto.

Eficiencia: Presenta la eficiencia de paralelización (speedup dividido por el número de hilos), lo que indica qué tan bien se aprovecha cada hilo adicional.

Puede cambiar entre las diferentes visualizaciones usando los botones en la parte superior del gráfico.

Conclusiones

1. Impacto de la política energética:

- La configuración energética tiene un impacto drástico en el rendimiento (hasta 181.6% de mejora)
- La transición de un perfil orientado a ahorro energético a uno balanceado proporciona mejoras significativas sin cambios de hardware

2. Arquitectura del procesador:

- El AMD Ryzen 7 7435HS con 8 núcleos supera significativamente al Intel Core i7-10510U de 4 núcleos
- La memoria más rápida (DDR5 vs DDR4) contribuye al mejor rendimiento del sistema AMD

3. Escalabilidad:

- La eficiencia del paralelismo está limitada por el número de núcleos físicos

- El AMD Ryzen 7 muestra mejor escalabilidad gracias a su mayor número de núcleos
 - Usar más hilos que núcleos lógicos disponibles no proporciona beneficios adicionales
4. **Tamaño del problema:**
- El rendimiento óptimo se logra con tamaños de matriz grandes (≥ 1600)
 - Las matrices pequeñas no aprovechan completamente el paralelismo disponible
5. **Consistencia:**
- La política energética afecta no solo al rendimiento sino también a la consistencia de los resultados
 - El perfil "balance_performance" proporciona resultados más predecibles que el perfil "power"

Recomendaciones

1. **Configuración del sistema:**
- Utilizar perfiles energéticos balanceados ("balance_performance" o "balance_power") en lugar de perfiles orientados exclusivamente al ahorro energético
 - Considerar el uso de sistemas con mayor número de núcleos para aplicaciones altamente paralelizables
2. **Optimización de código:**
- Ajustar el número de hilos al número de núcleos lógicos disponibles
 - Implementar técnicas de optimización de memoria como blocking/tiling para matrices grandes
 - Considerar la localidad de datos mediante el uso de matrices contiguas en memoria
3. **Metodología de pruebas:**
- Realizar múltiples iteraciones para obtener resultados estadísticamente significativos
 - Monitorear la temperatura del sistema para evitar efectos de throttling térmico

- Minimizar los procesos en segundo plano durante las pruebas de rendimiento

4. **Futuras investigaciones:**

- Evaluar el impacto de diferentes estrategias de scheduling en OpenMP (static, dynamic, guided)
- Investigar implementaciones basadas en SIMD (Single Instruction, Multiple Data)
- Explorar implementaciones híbridas OpenMP + MPI para sistemas distribuidos
- Comparar con implementaciones en GPU para problemas de gran escala

Este laboratorio demuestra la importancia crítica de considerar tanto el hardware como la configuración del sistema al optimizar aplicaciones de computación de alto rendimiento, y destaca cómo pequeños cambios en la configuración pueden tener impactos significativos en el rendimiento.

Referencias

1. OpenMP Architecture Review Board. (2021). OpenMP Application Programming Interface. <https://www.openmp.org/specifications/>
2. Buttari, A., Langou, J., Kurzak, J., & Dongarra, J. (2009). A class of parallel tiled linear algebra algorithms for multicore architectures. *Parallel Computing*, 35(1), 38-53.
3. Intel Corporation. (2023). Intel 64 and IA-32 Architectures Optimization Reference Manual.
4. AMD Corporation. (2024). Software Optimization Guide for AMD Family 19h Processors.
5. Walker, D. W. (2001). The hierarchical divide and conquer algorithm for parallel matrix multiplication. *Concurrency and Computation: Practice and Experience*, 13(8-9), 707-728