Scalability Analysis: The Ultimate Solver

Combining Tiling, OpenMP, and SIMD Vectorization

Progetto AMSC

December 2, 2025

Performance Overview: Combined Optimization (Float)

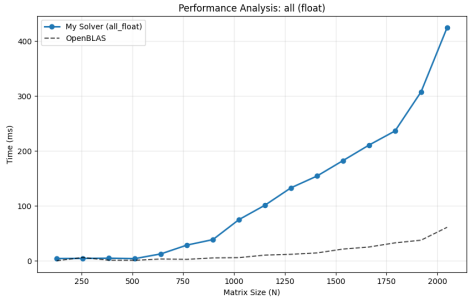

Figure: Execution Time (ms)

Linear scaling. Execution time for $N = 2048$ is just 0.36s.

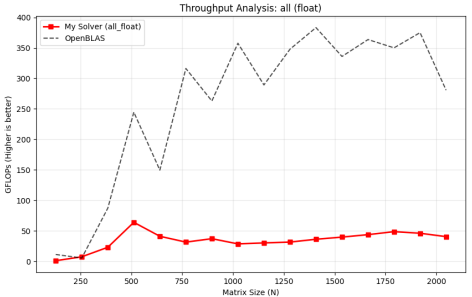

Figure: Throughput (GFLOPS)

Peak throughput reaches $\approx 47 - 50$ GFLOPS.

Quantitative Analysis: The Final Result

Final Benchmark Results ($N = 2048$)

Metric	Float	Double	Ratio
Time	**360 ms**	**815 ms**	2.2x
GFLOPS	47.66	21.06	BW Limit
vs Blas	7x Slower	7.3x Slower	Great!

Gain vs Tiling+OMP:
Float: $0.46s \rightarrow 0.36s$ (**+22% Faster**)
Explicit SIMD squeezes the last drops of performance.

Why is this the best?

This implementation attacks all three hardware bottlenecks simultaneously:

- **Latency:** Tiling fits data in L1/L2 Cache.
- **Throughput:** SIMD instructions compute 8 floats per cycle.
- **Concurrency:** OpenMP utilizes all CPU cores.

Project Summary: From Naive to Optimized

Evolution of Matrix Multiplication ($N = 2048$, Float)

Method	Strategy	Time (s)	Speedup
Naive	None (Baseline)	47.60 s	1x
OpenMP	Multithreading	14.39 s	3.3x
Tiling	Cache Blocking	1.32 s	36x
SIMD Unroll 2D	Vectorization	0.73 s	65x
Tiling + OpenMP	Blocking + Threads	0.46 s	103x
Final (All)	**Tiling + Threads + SIMD**	**0.36 s**	**132x**
OpenBLAS	*Assembly Tuned Library*	*0.05 s*	*950x*

Conclusion: By optimizing for Cache hierarchy, Vector registers, and Multi-core scaling, we reduced execution time by **two orders of magnitude**.