Parallel Matrix Multiplication Report

Shamil Davletov

October 2025

1 System specifications

- CPU model: Apple Silicon M1
- Cores: 8 cores (4 performance + 4 efficiency)
- Base frequency: 3.2 GHz
- RAM: 16 GB

2 Data tables

2.1 Strong scaling data

M	N	K	P	Tp	S	E	W	Cost	T0
300	300	300	1	0.199269	1.01543	1.01543	54000000	0.199269	0.0403207
300	300	300	2	0.099901	2.02544	1.01272	54000000	0.199802	0.0404285
300	300	300	3	0.06935	2.91771	0.972569	54000000	0.20805	0.0420975
300	300	300	4	0.125529	1.61192	0.402981	54000000	0.502115	0.101599
300	300	300	5	0.102464	1.97477	0.394954	54000000	0.51232	0.103664
300	300	300	6	0.0989618	2.04466	0.340776	54000000	0.593771	0.120145
300	300	300	7	0.0748909	2.70184	0.385977	54000000	0.524236	0.106075
300	300	300	8	0.0707181	2.86126	0.357658	54000000	0.565744	0.114474
300	300	300	9	0.0725641	2.78847	0.30983	54000000	0.653077	0.132146
300	300	300	10	0.0676482	2.99111	0.299111	54000000	0.676482	0.136881
300	300	300	11	0.0638139	3.17083	0.288257	54000000	0.701953	0.142035
300	300	300	12	0.0868952	2.32859	0.194049	54000000	1.04274	0.210992
300	300	300	13	0.0613279	3.29936	0.253797	54000000	0.797263	0.161321
300	300	300	14	0.058346	3.46798	0.247713	54000000	0.816844	0.165283
300	300	300	15	0.061342	3.2986	0.219907	54000000	0.92013	0.186182
300	300	300	16	0.056165	3.60265	0.225166	54000000	0.89864	0.181833

2.2 Weak scaling data

M	N	K	P	Tp	S	E	W	Cost	T0
300	300	300	1	0.197518	1.10273	1.10273	54000000	0.197518	0.0430212
377	377	377	2	0.194001	1.95707	0.978535	107165266	0.388002	0.147314
432	432	432	3	0.199119	2.89278	0.964259	161243136	0.597357	0.344082
476	476	476	4	0.245526	3.04194	0.760484	215700352	0.982104	0.733509
512	512	512	5	0.251391	4.13744	0.827489	268435456	1.25695	1.30738
545	545	545	6	0.257056	4.50658	0.751097	323757250	1.54234	1.78671
573	573	573	7	0.41375	3.22112	0.46016	376265034	2.89625	3.85995
600	600	600	8	0.412118	3.7593	0.469912	432000000	3.29694	5.10787
624	624	624	9	0.561674	3.10813	0.345347	485941248	5.05506	8.8249
646	646	646	10	0.649677	3.05929	0.305929	539172272	6.49677	12.9126
667	667	667	11	0.834934	2.55825	0.232569	593481926	9.18427	19.6174
686	686	686	12	0.742304	3.09973	0.258311	645657712	8.90765	20.496
705	705	705	13	0.711202	3.51099	0.270076	700805250	9.24562	23.0865
723	723	723	14	0.788632	3.41618	0.244013	755866134	11.0408	29.7453
739	739	739	15	0.905541	3.19859	0.21324	807166838	13.5831	39.3429
755	755	755	16	0.927175	3.31677	0.207298	860737750	14.8348	45.6203

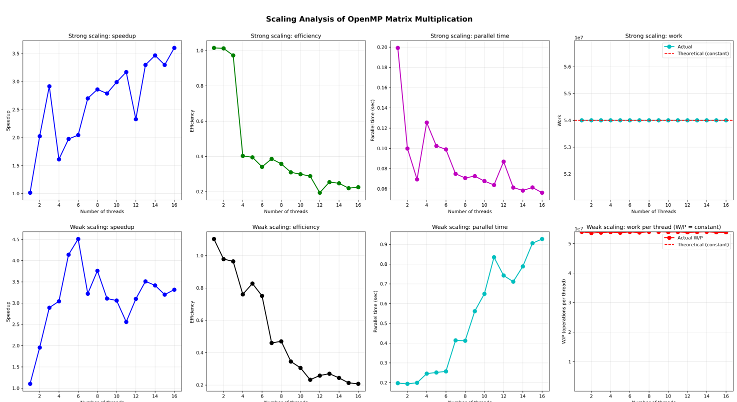
3 Computational Complexity

The naive matrix multiplication algorithm has complexity:

$$W = 2 * M * N * K$$

Each element requires n multiplications and n additions.

4 Scaling plots



5 Conclusions

- The best performance achieved with 6 threads in weak scaling. For fixed problem size speedup increases with more threads, showing effective parallelization
- Efficiency as we expected decreases with more threads due to increasing total overhead
- Parallel time decreases with increasing of threads
- Weak scaling maintains approximately constant work per thread as strong scaling with its constant work
- Performance degradation observed beyond 4 and 8 threads due to physical core limitations on M1 processor

6 Source code

GitHub repository:
`https://github.com/ninjasha/openmp-parallel-matrix-multiplication`