	MLOPS/s	Tile Size	Speed Up(%)	L1 Hit Rate	Matrix Size	Time to Solution(s)	L1 Miss Count	FLOPs	Note
Naive 1	840082671.6	NA	NA	0.4366819518	3000	64.279388	30424244465	54000000000	- Given that L3 cache's capacity is 36608KB, and each double is 8 bytes, an estimate size of matrix for multiplication that exceeds L3 cache is ~2200x2200. - Total Floating Point Operations would be 2N^3 (N^2 elements, each element requires N multiplications and ~N additions, where N is matrix row/column). Thus, MLOP/s can be calculated using (2N^3)/TIME_TO_SOLUTION - We can use Matrix Size and L1 Miss Count to estimate L1 Hit Rate. Each iteration in the most inner for loop includes 1 memory read for each elements in x and y (total 2), and there is 1 memory write to matrix z. Therefore, for NxN matrices, we have 2N^3 memory reads and N^2 memory write. Therefore, total memory access count would be 2N^3 + N^2. Then L1 Hit Rate = 1 - (L1_MISS_COUNT / MEMORY_ACCESS_COUNT) - Each program is compiled with -O2 flag
Tiled	2277707649	16	0.6311718618	0.9831098961	3000	23.708047	912217621	54000000000	
Tiled	2173597329	32	0.6135058411	0.9946230213	3000	24.843608	290405241	54000000000	
Tiled	2155886482	64	0.6103307486	0.9796836558	3000	25.047701	1097265436	54000000000	
Naive 2	1013572094	NA	NA	0.4361703974	2300	24.008159	13723212208	24334000000	
Tiled	2310903840	16	0.5613958155	0.9814357321	2300	10.530079	451841100	24334000000	
Tiled	2221529205	32	0.5437502726	0.9941906756	2300	10.953716	141394832	24334000000	
Tiled	2147252229	64	0.5279678463	0.9555519702	2300	11.332623	1081833487	24334000000	
Naive 3	1363324658	NA	NA	0.4352087836	1000	1.467002	1130147224	2000000000	
Tiled	2502749896	16	0.455269318	0.9843116082	1000	0.799121	31392472	2000000000	
Tiled	2274782332	32	0.400679072	0.9946911854	1000	0.879205	10622938	2000000000	
Tiled	2140589647	64	0.3631078894	0.9820469045	1000	0.934322	35924144	2000000000	

Running the above experiments show indications that higher L1 Hit Rate indicates faster time to solution, but not always, which could be due to other system constraints or mechanisms. I also observed that the optimal tile size for matrix multiplication seems to be around ~16. Running programs with a tile size = 8 increases time execution time, which is not shown in this chart.