CS 575 Project 4

Jeremy Dao

May 2020

1. The following results were gathered using a machine with a Intel i7-7700k CPU (4 cores, 8 threads) running at 4.20GHz.

2. The following table displays the SIMD vs. Non-SIMD performance for varying array sizes. Note that all performance values are in units of MegaMultiplies per Second.

Array Size	SIMD Performance	Non-SIMD Performance	Speedup
1024	219.27	87.99	2.49
2048	1226.34	486.00	2.52
4096	1239.72	487.04	2.55
8192	1245.17	486.69	2.56
16384	1248.78	487.13	2.56
24576	1249.80	487.32	2.56
35840	1247.87	486.99	2.56
49152	1247.29	486.95	2.56
65536	1249.71	486.85	2.57
102400	1247.50	486.75	2.56
204800	1274.50	497.94	2.56
307200	1246.31	486.91	2.56
409600	1274.54	497.98	2.56
512000	1272.63	497.67	2.56
614400	1269.93	496.94	2.56
716800	1264.40	496.27	2.55
819200	1246.77	494.06	2.52
1048576	1210.95	490.30	2.47
2097152	1172.60	480.22	2.44
4194304	1161.75	478.47	2.43
6291456	1164.13	480.08	2.42
8388608	1160.25	479.34	2.42

3. The following graph displays the speedup data from the table above.

4. We can see from the graph there is an initial peak in speedup before speedup plateaus back down again. The initial peak is caused by the low speedup value for small array size. In this case, the array size is too small to overcome the SIMD overhead of splitting the array and using the different registers, and thus speedup is lower. Speedup then peaks around 2.56 before dipping back down for very large array sizes. This could be caused by large array sizes affecting the prefetcher and possibly because it is not expecting an array so large, it does not prefetch far ahead enough.

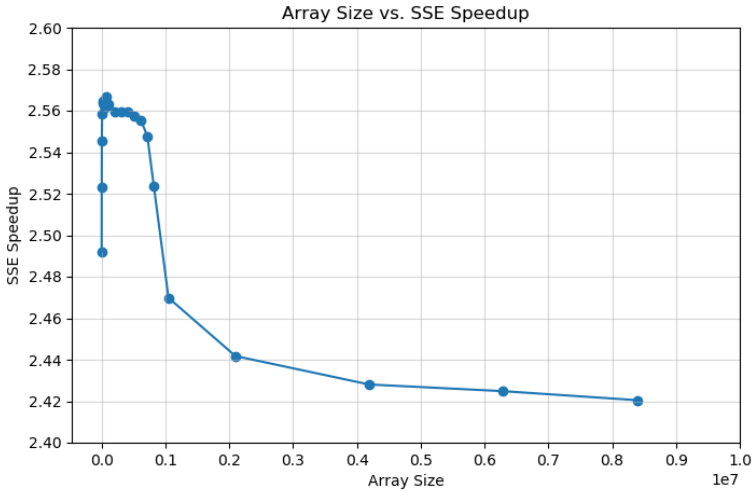

Figure 1: SIMD speed up vs. Array size

5. The following tables show the results for adding multi-threading to SIMD. The first table shows the SIMD performance for varying array sizes and number of threads:

Array Size	1 Thread	2 Threads	4 Threads
1024	148.79	203.30	180.44
2048	972.00	416.01	1503.67
4096	1089.65	627.07	2203.33
8192	1160.01	777.45	2993.08
16384	1203.73	1061.07	3657.13
24576	1220.38	1303.15	4284.53
35840	1223.17	1560.36	3760.76
49152	1229.84	1807.26	4532.23
65536	1263.59	2036.23	4685.82
102400	1266.54	2440.54	4806.84
204800	1246.44	2478.10	4890.05
307200	1248.02	2481.78	4931.53
409600	1251.51	2480.08	4933.45
512000	1273.39	2481.74	4930.09
614400	1271.04	2477.83	4936.05
716800	1263.09	2456.33	4911.17
819200	1256.87	2449.40	4875.52
1048576	1206.65	2397.00	4754.61
2097152	1175.12	2275.36	4311.95
4194304	1168.05	2239.79	3962.02
6291456	1161.61	2217.90	3603.35
8388608	1160.42	2211.06	3538.31

The next table shows the Non-SIMD performance for varying array sizes and number of threads:

Array Size	1 Thread	2 Threads	4 Threads
1024	73.49	128.90	131.64
2048	438.26	208.05	934.30
4096	460.64	308.88	1049.72
8192	472.27	339.40	1356.74
16384	479.05	441.24	1464.17
24576	481.82	533.44	1824.64
35840	482.53	626.45	1420.03
49152	483.66	716.64	1871.39
65536	495.39	806.89	1898.82
102400	495.88	965.18	1915.67
204800	485.69	968.36	1929.11
307200	486.00	969.78	1935.24
409600	497.09	970.40	1936.08
512000	497.20	970.95	1936.18
614400	497.13	970.35	1938.37
716800	496.15	968.56	1935.17
819200	495.10	967.77	1933.21
1048576	489.40	959.94	1919.60
2097152	480.39	943.79	1889.95
4194304	479.81	942.35	1883.49
6291456	479.84	940.33	1872.96
8388608	479.98	937.64	1869.25

The last table shows the speedup gained going from Non-SIMD to SIMD for varying array sizes and threads. Note that data for 1 thread is not shown since that is the baseline, and thus is 1 for all array sizes.

Array Size	2 Threads	4 Threads
1024	2.77	2.46
2048	0.95	3.43
4096	1.36	4.78
8192	1.65	6.34
16384	2.21	7.63
24576	2.70	8.89
35840	3.23	7.79
49152	3.74	9.37
65536	4.11	9.46
102400	4.92	9.69
204800	5.10	10.07
307200	5.11	10.15
409600	4.99	9.92
512000	4.99	9.92
614400	4.98	9.93
716800	4.95	9.90
819200	4.95	9.85
1048576	4.90	9.72
2097152	4.74	8.98
4194304	4.67	8.26
6291456	4.62	7.51
8388608	4.61	7.37

6. The following graph displays all the speedup data from the above tables into a single graph, showing the speed up gained when using a vary number of threads with and without SIMD:

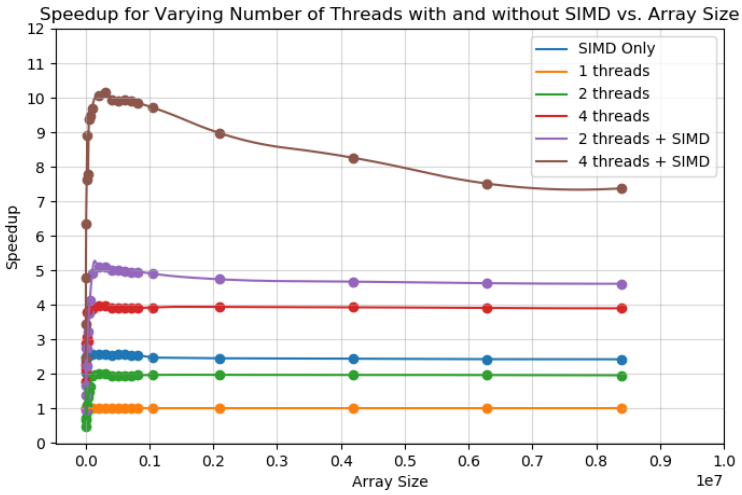

Figure 2: Speedup vs. Array size for varying number of threads with and without SIMD

We can see that for 2 and 4 threads alone, we get the expected 2x 4x speedup, along with the usual low speedup behavior for small array sizes due to the overhead of parallelization outweighing the speedup benefit when there is such little data. It is clear than adding SIMD along with multiple threads drastically improves performance, achieving much higher speedups can just multithreading along. Furthermore, for the multithread + SIMD cases we can see the downward curve, similar to what we saw in the single thread SIMD graph above, that could be caused by non-optimal prefetching. This is seems to have a larger affect on the 4 threads case than the 2 threads case since there are more threads to prefetch for, exacerbating the issue.