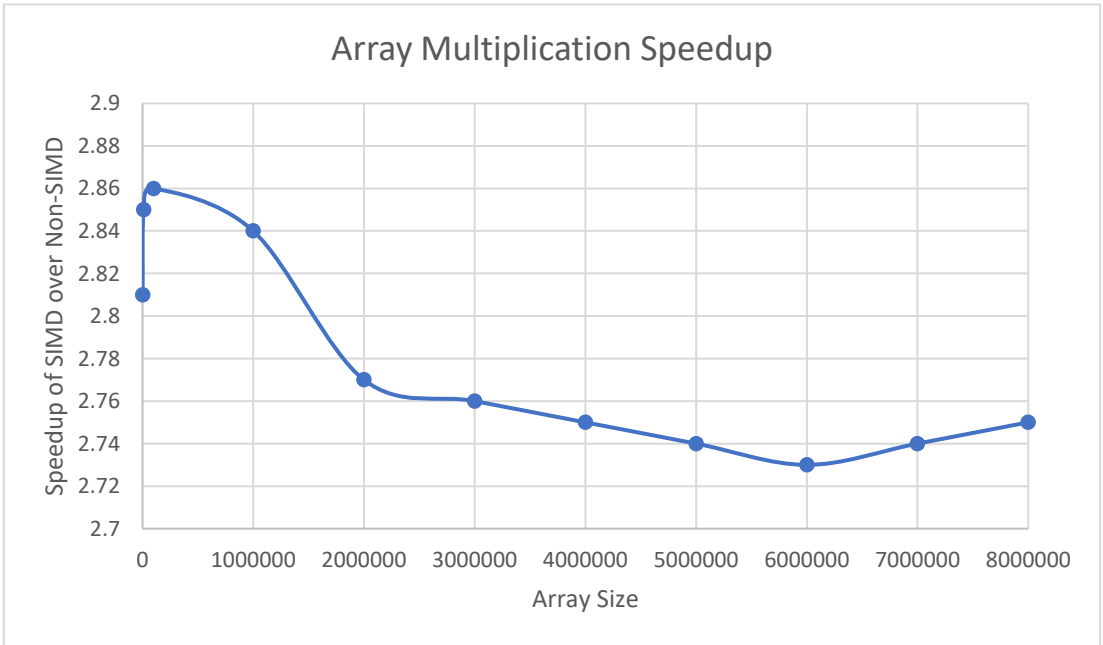
Xiaoying Li
lixiaoyi@oregonstate.edu
Project #4 Vectorized Array Multiplication/Reduction using SSE

- **Machine:**
  I ran the program on OSU Engr Server flip3 (without -O3).

- **Table of performances for each array size and the corresponding speedups:**

Array Size	Non-SIMD Performance	SIMD Performance	Speedup
1000	223.56	627.37	2.81
10000	225.89	643.71	2.85
100000	224.85	643.43	2.86
1000000	222.49	630.99	2.84
2000000	220.27	610.62	2.77
3000000	220.09	606.41	2.76
4000000	219.73	604.84	2.75
5000000	219.45	600.35	2.74
6000000	218.82	597.42	2.73
7000000	219.76	603.05	2.74
8000000	219.66	604.21	2.75

- **Graph of SIMD/non-SIMD speedup versus array size:**



- **What patterns are you seeing in the speedups? Are they consistent across a variety of array sizes? Why or why not, do you think?**

The speedups range from 2.7 ~ 2.9. And when the array sizes are not that big (less than 1M), the speedup rises as the array size gets bigger. But after that peak, when the array size gets bigger, the speedup drops.

In general, the speedups are consistent across a variety of array sizes. Because SSE allows four floating point operations to happen simultaneously, and the speedup brought by this won't change much as the array size changes. For the little better speedup under small array sizes, I think it's because the Intrinsics have a tighter coupling to the setting up of the registers, and a smaller setup time makes the small array size speedup look better. And for the little drops under big array sizes, I think the reason is the temporal cache problem. Because we only use the values once, and the speed of multiplication gets ahead of the ability of bringing memory in from cache. But in general, the speedups not change much under different array sizes.

***** Extra Credit *****

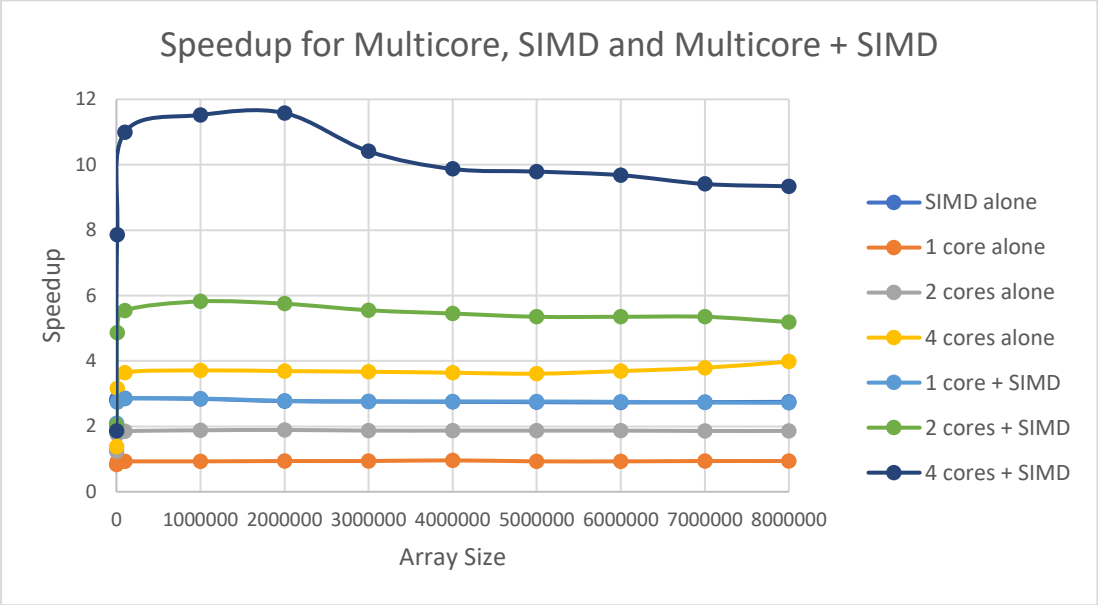
- **Table of performances:**

Array Size	No Multicore or SIMD Performance	1 Core Alone Performance	Speedup
1000	223.72	186.51	0.83
10000	225.96	207.18	0.92
100000	223.84	208.85	0.93
1000000	220.83	205.89	0.93
2000000	215.38	202.45	0.94
3000000	221.49	207.09	0.94
4000000	208.63	200.75	0.96
5000000	212.39	198.32	0.93
6000000	217.42	202.88	0.93
7000000	217.52	204.21	0.94
8000000	215.37	201.63	0.94
Array Size	No Multicore or SIMD Performance	2 Cores Alone Performance	Speedup
1000	223.72	277.38	1.24
10000	225.91	396.62	1.76
100000	224.83	415.51	1.85
1000000	218.99	412.09	1.88
2000000	213.68	404.57	1.89
3000000	218.39	408.64	1.87
4000000	219.58	409.21	1.86
5000000	216.76	404.56	1.87
6000000	215.51	404.23	1.88
7000000	219.04	407.58	1.86

8000000	216.44	407.06	1.88
Array Size	No Multicore or SIMD Performance	4 Cores Alone Performance	Speedup
1000	223.72	308.64	1.38
10000	225.86	714.14	3.16
100000	222.52	810.01	3.64
1000000	217.63	808.16	3.71
2000000	216.85	802.19	3.69
3000000	217.49	798.27	3.67
4000000	217.86	793.37	3.64
5000000	215.33	775.35	3.61
6000000	213.75	790.31	3.69
7000000	212.94	807.31	3.79
8000000	198.51	790.05	3.98
Array Size	No Multicore or SIMD Performance	1 Core SIMD Performance	Speedup
1000	223.72	468.88	2.09
10000	225.88	618.66	2.74
100000	224.05	637.85	2.85
1000000	219.27	625.25	2.85
2000000	226.01	628.29	2.78
3000000	226.45	623.95	2.76
4000000	221.85	611.34	2.76
5000000	217.99	600.76	2.76
6000000	225.91	620.46	2.75
7000000	225.56	616.07	2.73
8000000	227.94	620.97	2.72
Array Size	No Multicore or SIMD Performance	2 Cores SIMD Performance	Speedup
1000	233.75	474.89	2.03
10000	225.91	1098.63	4.86
100000	224.98	1245.29	5.54
1000000	218.12	1269.43	5.82
2000000	218.59	1256.81	5.75
3000000	170.19	944.35	5.55
4000000	217.27	1184.05	5.45
5000000	219.88	1177.23	5.35
6000000	219.24	1173.39	5.35
7000000	218.34	1168.06	5.35
8000000	226.34	1175.69	5.19
Array Size	No Multicore or SIMD Performance	4 Cores SIMD Performance	Speedup
1000	223.72	415.94	1.86

10000	225.88	1775.51	7.86
100000	132.96	1461.77	10.99
1000000	221.26	2548.14	11.52
2000000	215.92	2500.11	11.58
3000000	216.54	2254.09	10.41
4000000	226.85	2238.88	9.87
5000000	219.63	2151.61	9.79
6000000	220.99	2139.33	9.68
7000000	226.19	2126.29	9.41
8000000	226.21	2112.63	9.34

- **Graph of performances:**



- **What the curves are showing and why?**

The curves show that combine SIMD and multicore can get large speedups, which are much better than both SIMD alone and multicores alone. The speedup under multicore + SIMD is almost the product of speedup under same SIMD alone and multicore alone. For example, the speedup under SIMD alone is about 2.7 ~ 2.9, and the speedup under 4 cores alone is about 3 ~ 4, then the speedup under 4 cores + SIMD is about 9 ~ 12.

The reason is SIMD allows arithmetic operations on multiple array elements to be performed simultaneously, and multicore distributes these operations to different threads, which combines their speedups together. Thus, the speedup under multicore + SIMD is almost the product of speedup under same SIMD alone and multicore alone. And since there are always some setting up time, the speedup cannot reach the ideal result.