

Jung-Che Chang

Project #4

Vectorized Array Multiplication and  
Multiplication/Reduction using SSE

[changju@oregonstate.edu](mailto:changju@oregonstate.edu)

**1. What machine you ran this on**

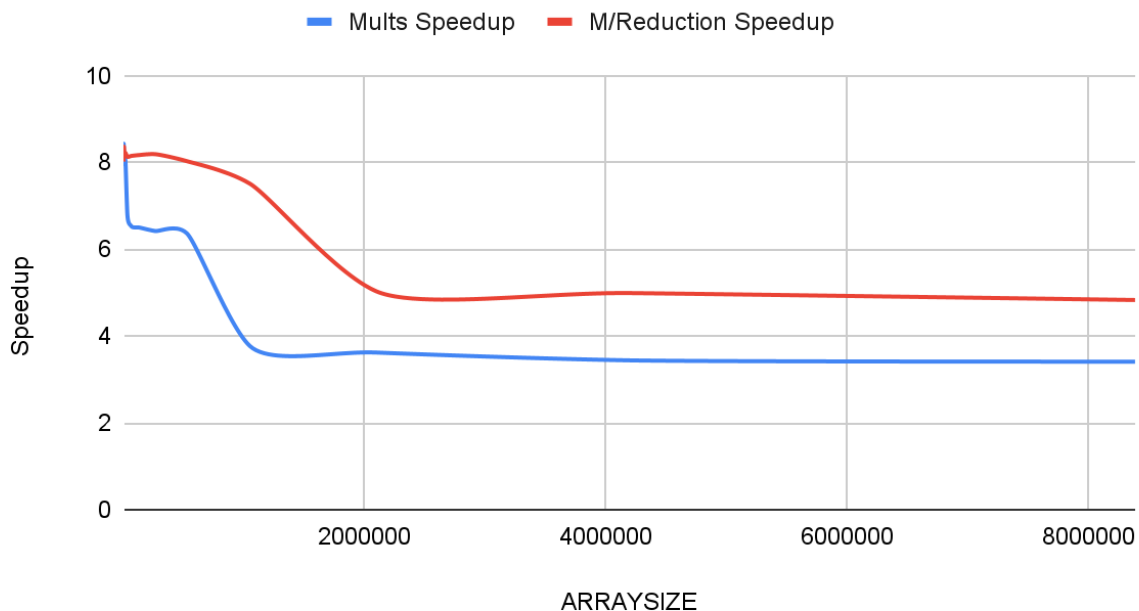
flip2

**2. Show the 2 tables of performances for each array size and the corresponding speedups**

ARRAY SIZE	Non-SIM D Mults	SIMD Mults	Mults Speedup	Non-SIMD M/Reduction	SIMD M/Reduction	M/Reduction Speedup
1024	120.89	990.33	8.19	121.86	977.13	8.02
2048	121.27	1021.97	8.43	122.54	1017.95	8.31
4096	121.25	967.61	7.98	122.86	1027.64	8.36
8192	120.81	972.45	8.05	123.18	1011.63	8.21
16384	120.96	972.63	8.04	123.24	1013.23	8.22
32768	120.78	820.02	6.79	123.26	1001.68	8.13
65536	120.78	788.22	6.53	123.24	1004.68	8.15
131072	120.3	781.84	6.5	122.6	1002.22	8.17
262144	119.85	769.74	6.42	122.51	1003	8.19
524288	119.48	760.56	6.37	232.01	1863.91	8.03
1048576	217.18	818.84	3.77	227.21	1707.31	7.51
2097152	207.97	752.46	3.62	222.35	1118.92	5.03
4194304	218.38	751.29	3.44	228.37	1139.37	4.99
8388608	220.79	751.91	3.41	225.57	1230.59	4.83

**3. Show the graphs (or graph) of SIMD/non-SIMD speedup versus array size**

**SIMD/NON SIMD speedup versus array size**



**4. What patterns are you seeing in the speedups?**

From this graph, we can see that as the array size grows, the general trend of speedup is decreasing.

**5. Are they consistent across a variety of array sizes?**

No, they are not consistent with different array sizes.

**6. Why or why not, do you think?**

I think this is because of the limitations of memory transfer speed. SIMD processing is efficient for small array sizes, it can utilize the cache line effectively and provide a speed boost. However, as the array size increases, the next cache line retrieval becomes delayed, while a larger number of cache lines is required to process the data. Consequently, this leads to a decrease in the overall speedup.