Vectorized Array Multiplication/Reduction using SSE

Rui Gao

gaorui@oregonstate.edu

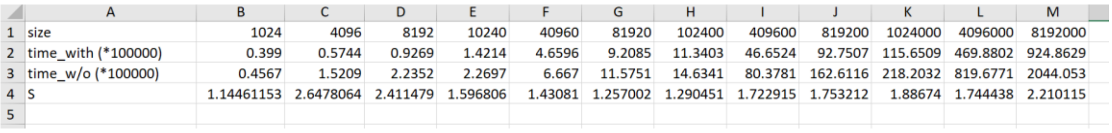
Project 4

CS 575

1. What machine you ran this on

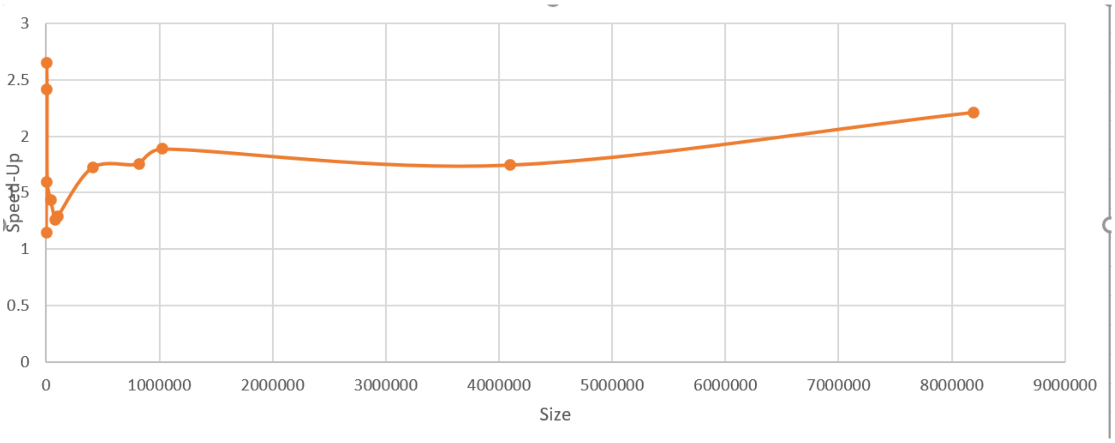
   Zephyrus 14

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



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	size	1024	4096	8192	10240	40960	81920	102400	409600	819200	1024000	4096000	8192000
2	time_with (*100000)	0.399	0.5744	0.9269	1.4214	4.6596	9.2085	11.3403	46.6524	92.7507	115.6509	469.8802	924.8629
3	time_w/o (*100000)	0.4567	1.5209	2.2352	2.2697	6.667	11.5751	14.6341	80.3781	162.6116	218.2032	819.6771	2044.053
4	S	1.14461153	2.6478064	2.411479	1.596806	1.43081	1.257002	1.290451	1.722915	1.753212	1.88674	1.744438	2.210115
5													

3. Show the graph of SIMD/non-SIMD speedup versus array size (one curve only)



4. What patterns are you seeing in the speedups?

The curve increased first and then decreased. In the end, it increased again.

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

No, they are not consistent. It has fluctuation.

6. Why or why not, do you think?

In the beginning, the size is small, so the real computation time occupies a small part of time and so, the S is small.

Then, as the size increase, so the real computation time occupies a bigger and bigger part of time and so, the advantage of SSE became obvious and the trending showed increasing.

Next, maybe the time costing of assigning increased, so the S decreased.

In the end, I don’t know, it backs to normal.