Deqing Qu
qud@oregonstate.edu

1. Tell what machine you ran this on

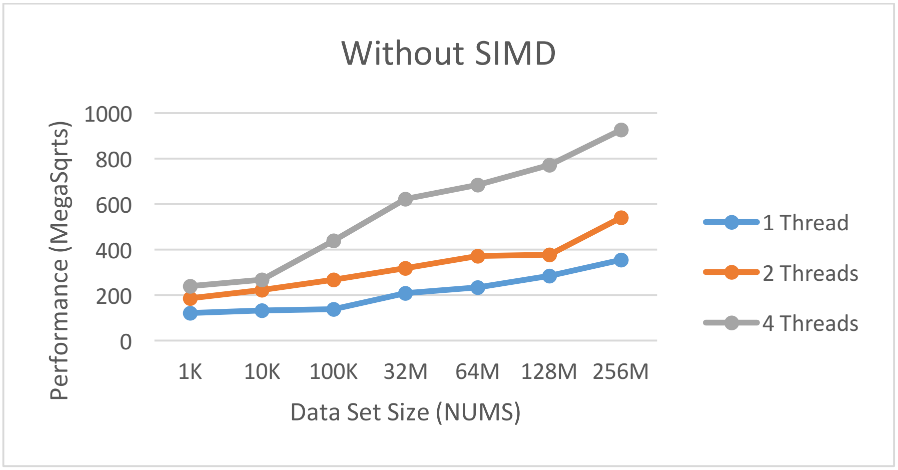
   I ran the code on rabbit.

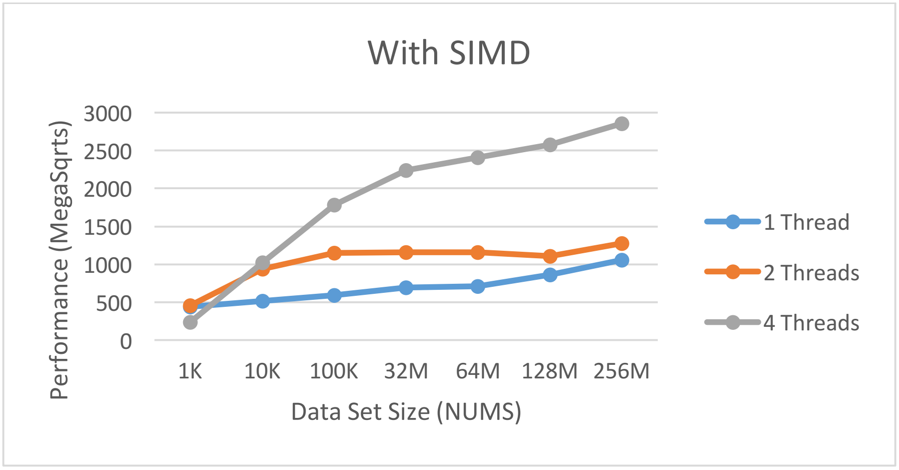
2. Create a table with your results

NUMT \ NUMS	1K	10K	100K	32M	64M	128M	256M
1(NON-VEC)	121.56	132.65	140.21	209.74	235.56	284.76	356.49
2(NON-VEC)	188.07	222.35	267.58	318.49	372.01	378.24	542.61
4(NON-VEC)	239.01	268.72	440.33	624.02	683.82	772.39	927.27
1	442.12	516.38	594.6	695.24	710.46	865.17	1062.51
2	457.23	937.9	1148.89	1159.87	1160.03	1105.07	1275.17
4	236.68	1023.56	1784.46	2236.66	2409.53	2577.42	2855.6

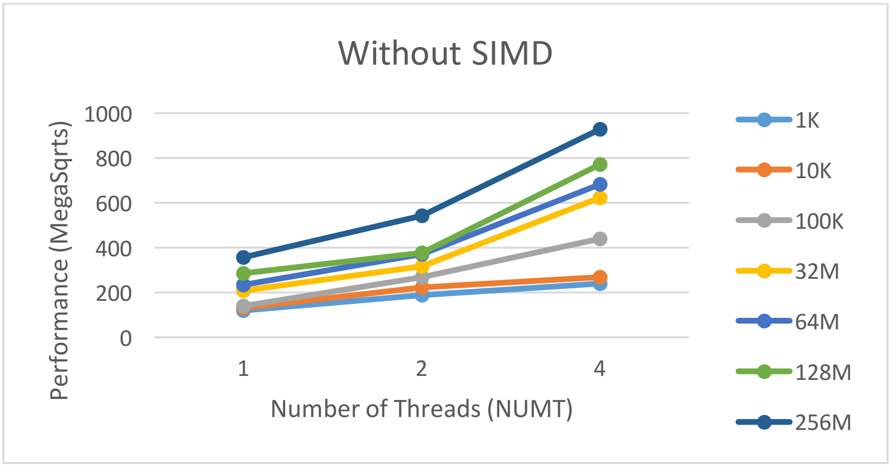
   The first row shows the size of data set, from 1K to 256M. The first column indicates the number of threads (with or without vectorized), from 1 to 4.

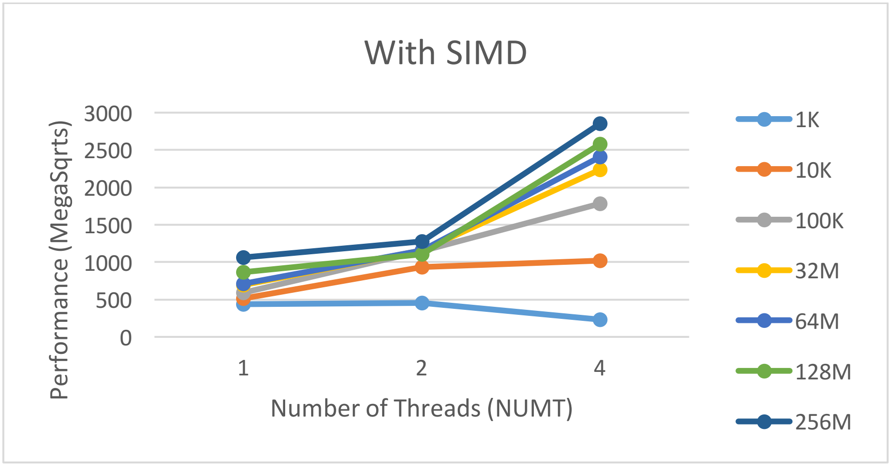
3. Draw graphs

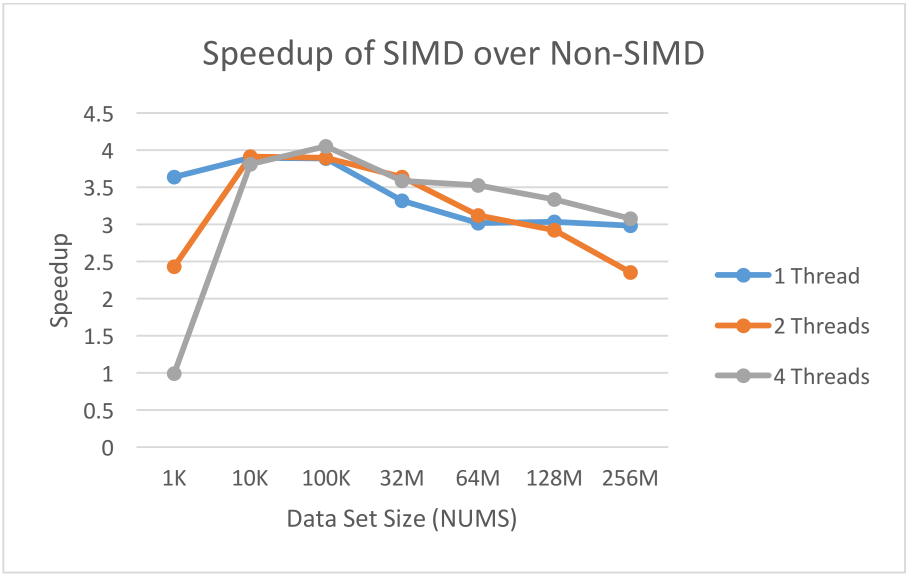




Deqing Qu
qud@oregonstate.edu







4. What patterns are you seeing in the performance curves?

Deqing Qu
qud@oregonstate.edu

(1) When the size of data set is very small, the speedup of SIMD is not as much as expected.
(2) The speedup is up to 4.
(3) The speedup of SIMD over non-SIMD will drop when the size of data set is more than 32M.

5. Why do you think the patterns look this way?

(1) If the data size is too small, the benefit of SIMD will be shaded by the additional work for SIMD.
(2) I guess the processor of RABBIT supports for the 4-way SIMD. So the maximum speedup of SIMD over non-SIMD is 4.
(3) The temporal coherence is violated when the size of data set is too much. The data in cache line is possibly only used once and then be replaced.

6. What does that mean for the proper use of vectorized parallel programming?

The data set size is essential for performance. If the dataset size is too small, the benefit of SIMD will be shaded by the additional work for SIMD. If the dataset size is too much, the temporal coherence is violated and the performance will decrease.