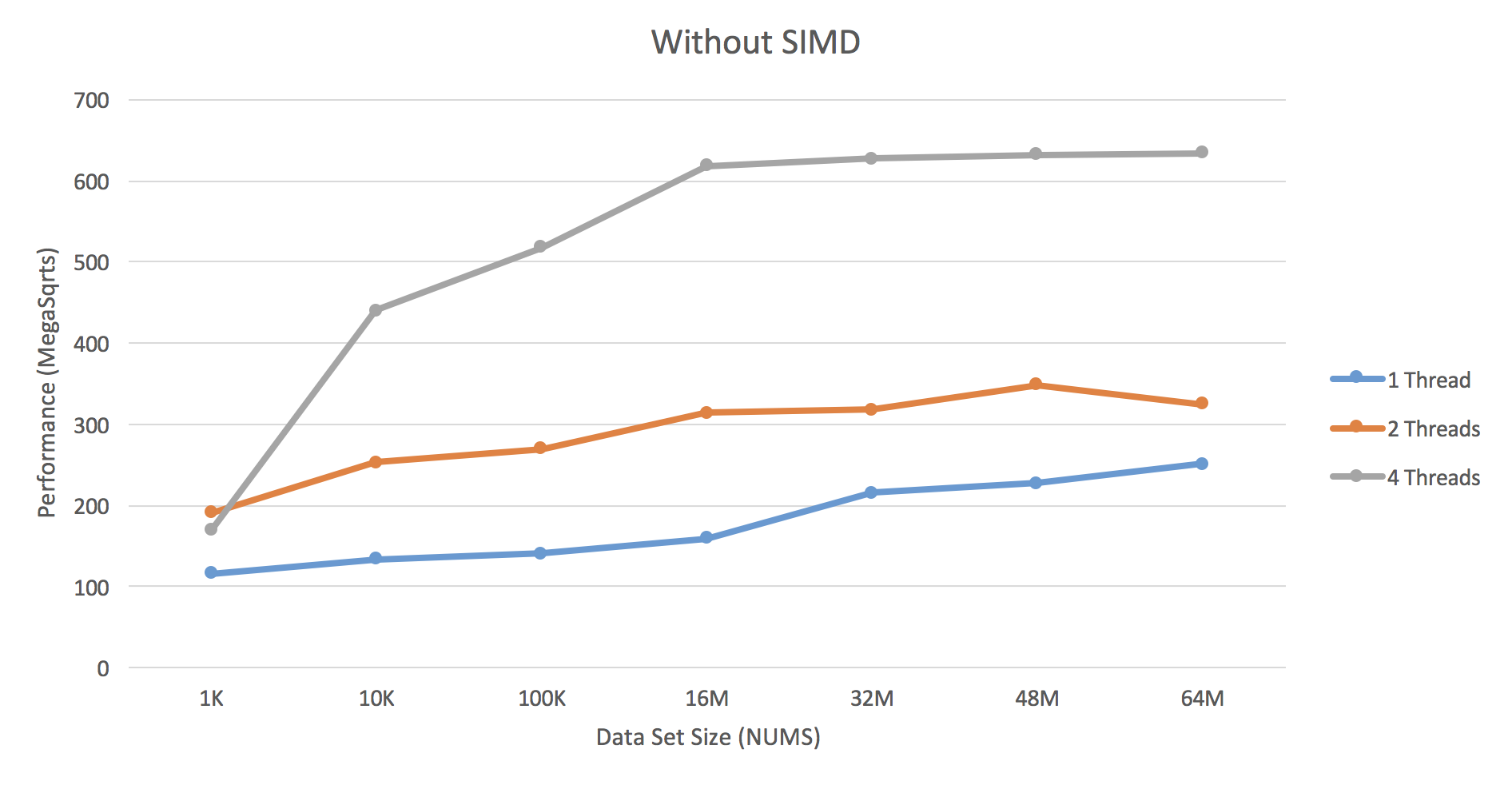
1. Tell what machine you ran this on

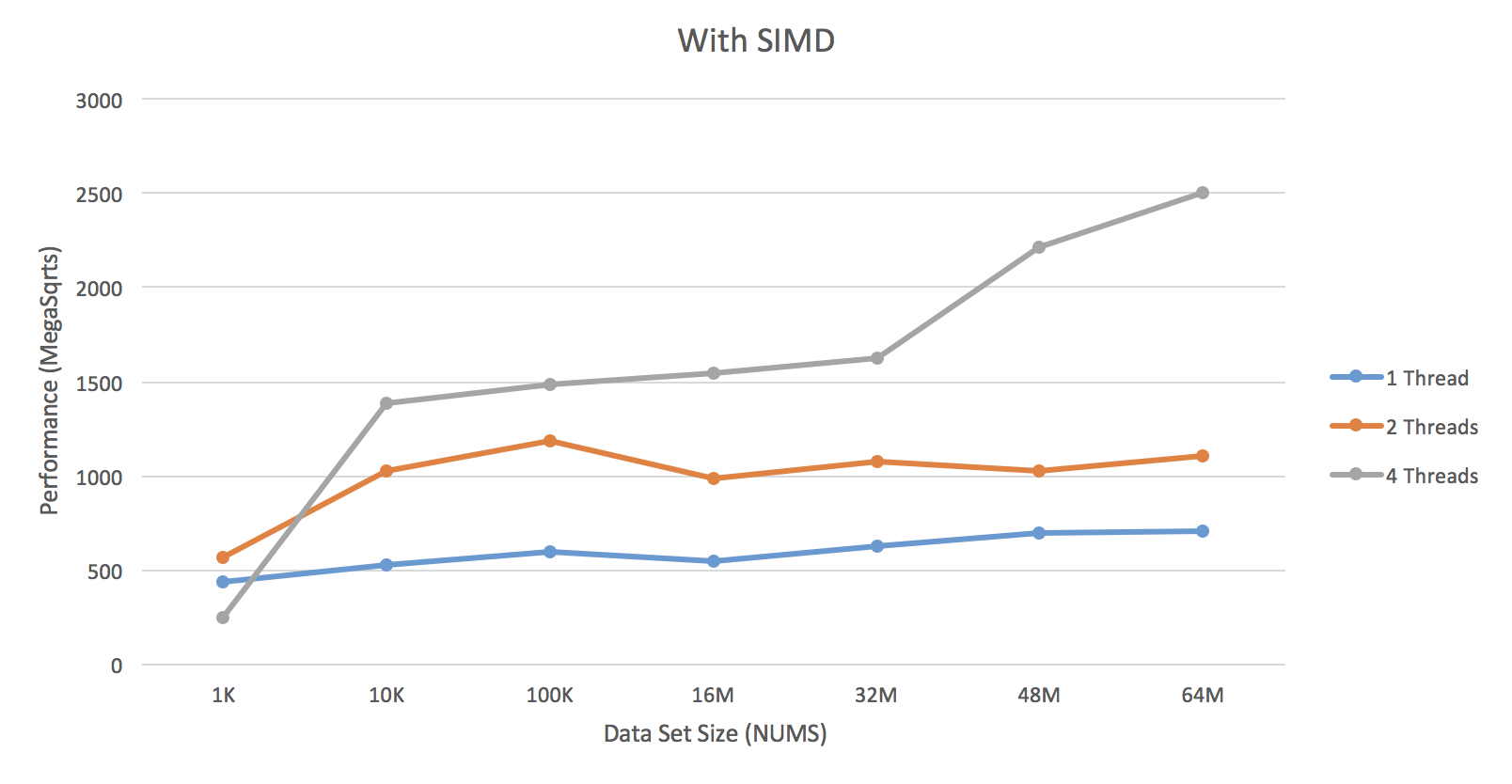
I ran the code on rabbit.

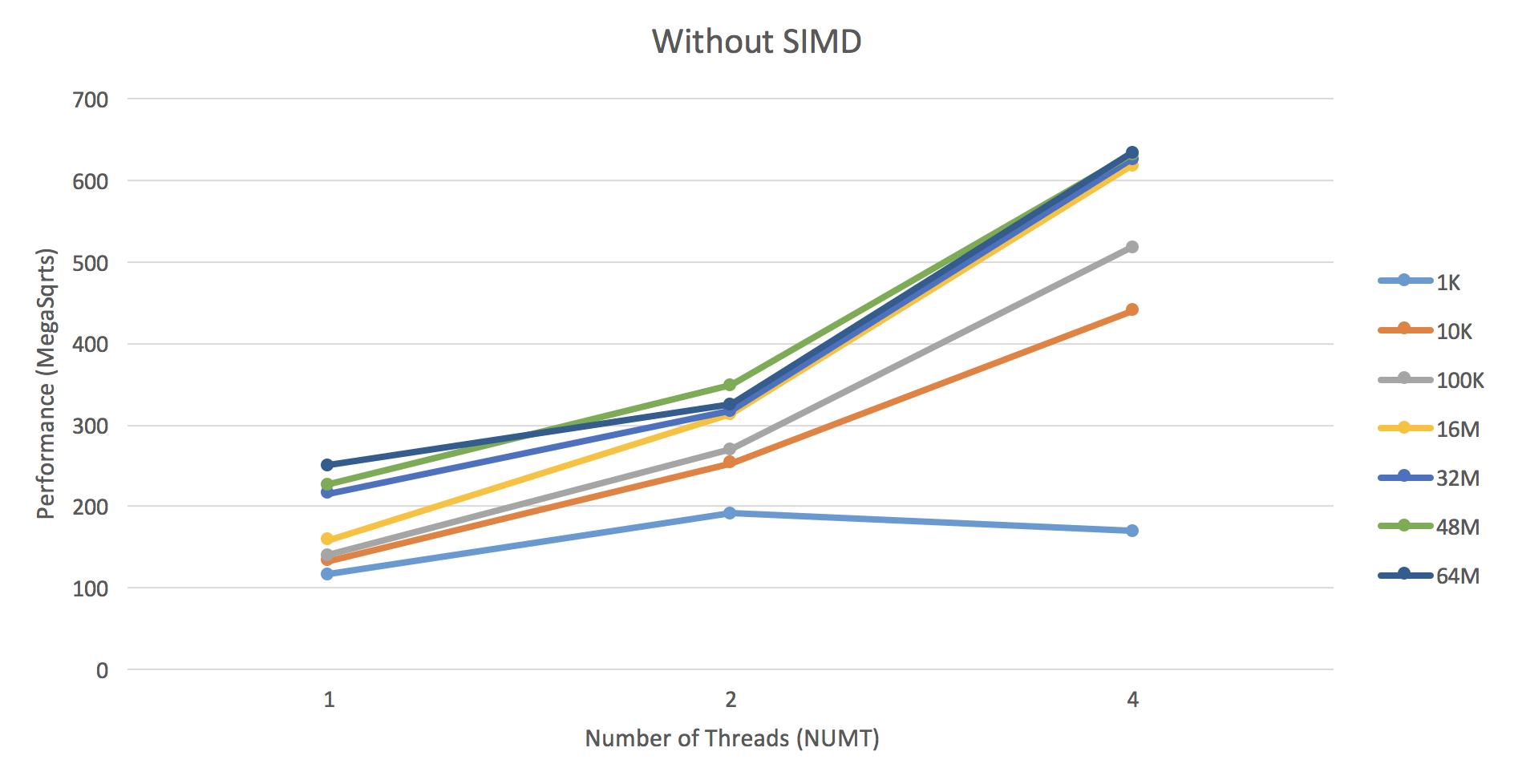
1. Create a table with your results

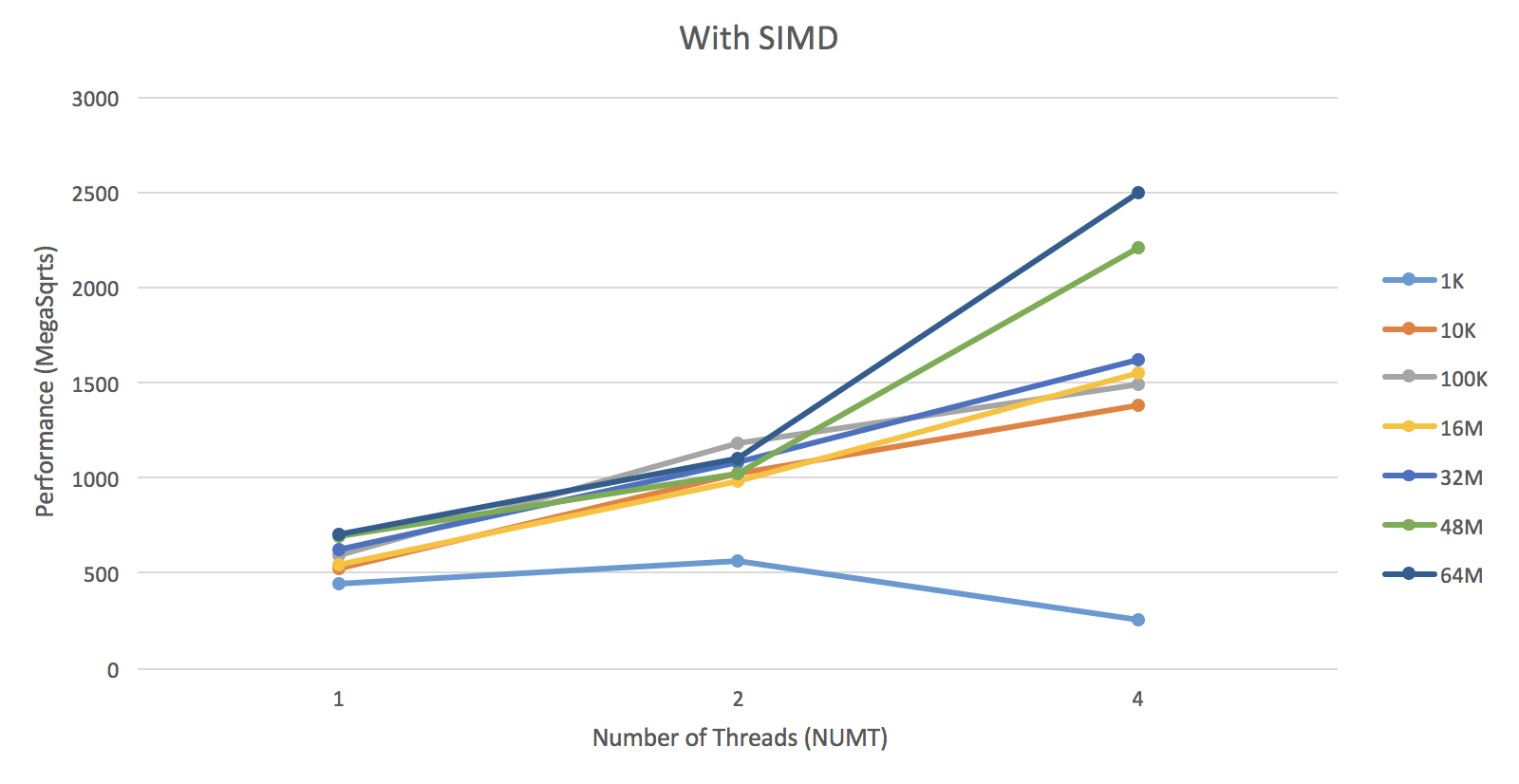
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| NUMT | 1K | 10K | 100K | 16M | 32M | 48M | 64M |
| 1(NON-VEC) | 116.19 | 133.74 | 140.51 | 159.28 | 215.84 | 227.14 | 250.8 |
| 2(NON-VEC) | 191.16 | 252.93 | 269.82 | 313.88 | 317.88 | 348.96 | 324.85 |
| 4(NON-VEC) | 169.75 | 440.6 | 518.13 | 618.56 | 626.81 | 631.88 | 634.36 |
| 1 | 442.12 | 528.62 | 596.49 | 544.84 | 628.49 | 696.9 | 704.72 |
| 2 | 565.16 | 1023.75 | 1181.69 | 983.61 | 1078.86 | 1025.53 | 1105.4 |
| 4 | 252.49 | 1382.07 | 1489.31 | 1548.02 | 1625.46 | 2210.8 | 2499.51 |

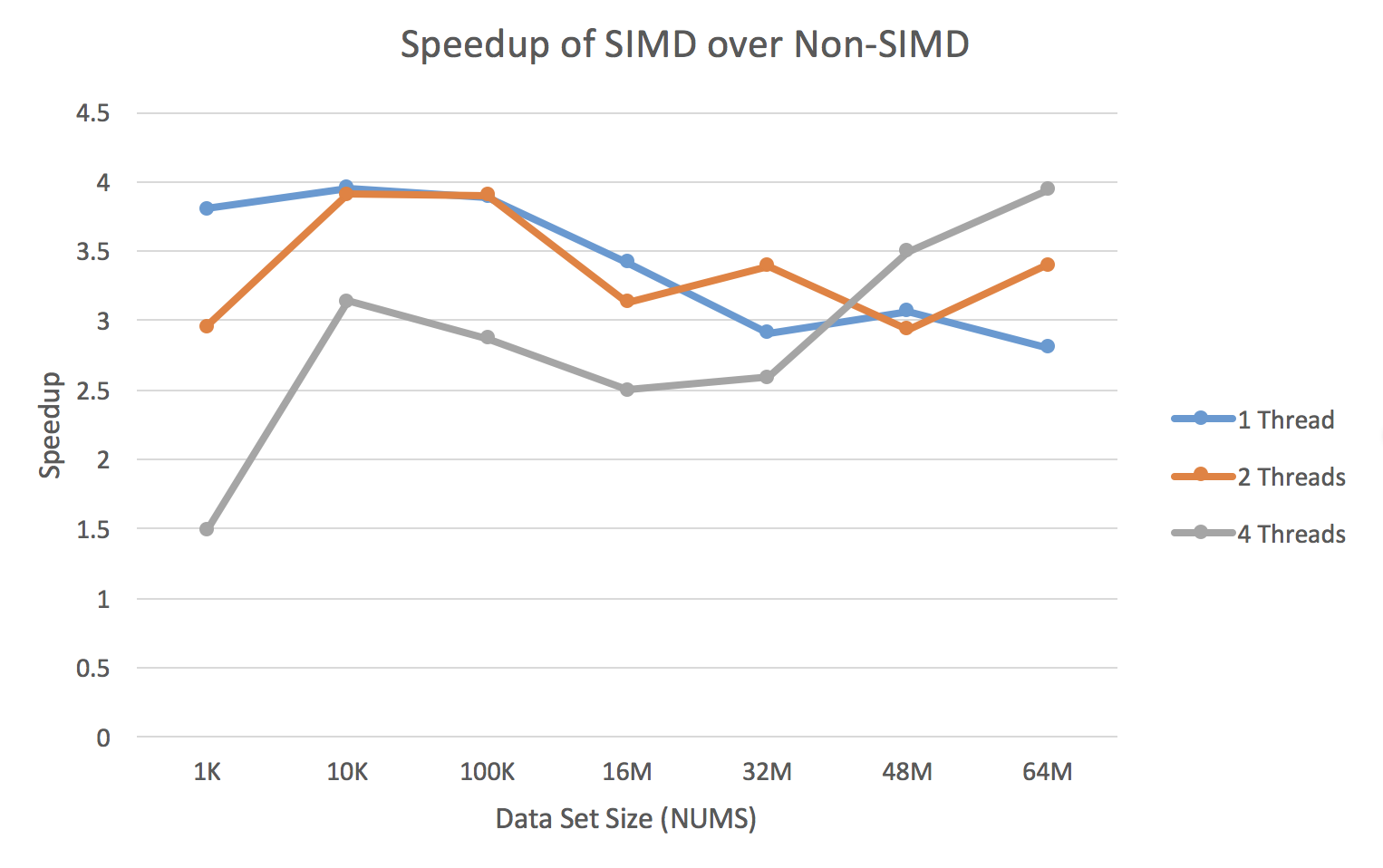
1. Draw graphs











1. What patterns are you seeing in the performance curves?
   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. As mentioned in materials, the speedup of SIMD over non-SIMD will drop when the size of data set is too much. But it did not happen in my experiments.
2. 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 of 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.
3. 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.