

Kristin Schaefer

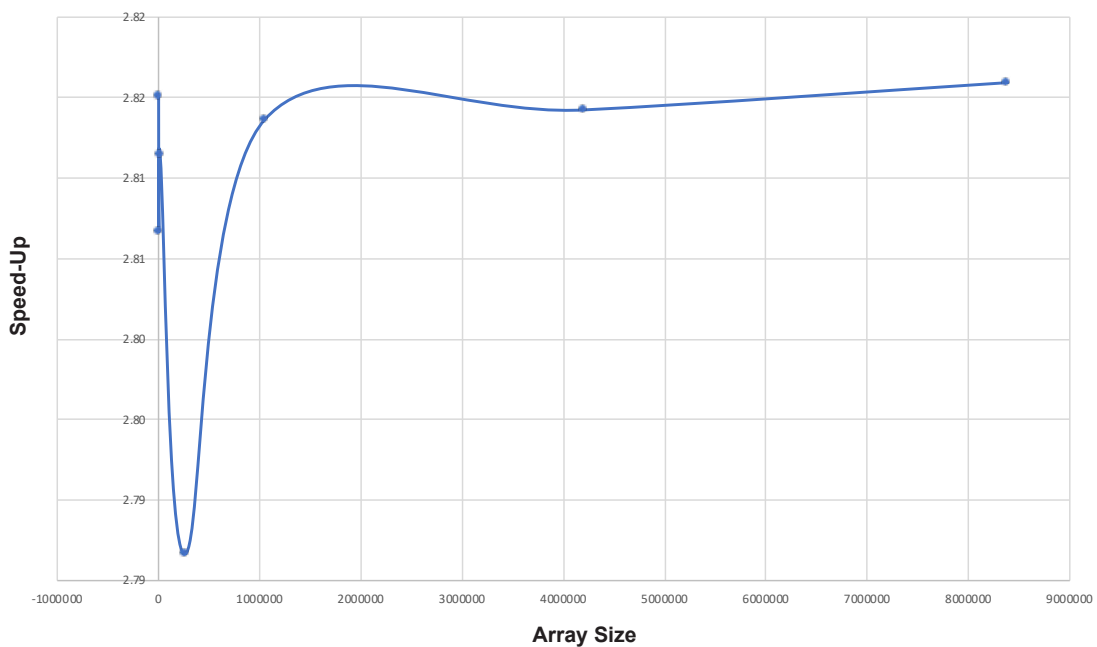
Project #4 : Vectorized Array Multiplication/Reduction Using SSE

Machine: OSU flip (Linux)

Array Size vs. Speed-Up

Array Size	SIMD Speedup	SIMD Performance MegaMults/Sec	Non-SIMD Performance MegaMults/Sec
1024	2.82	658.00	233.74
4096	2.81	656.03	233.74
16384	2.81	628.29	223.48
262144	2.79	618.05	221.79
1048576	2.81	628.65	223.43
4194304	2.81	657.80	233.74
8388608	2.82	658.00	233.67

Array Size vs. Speed-Up



Kristin Schaefer

Project #4 : Vectorized Array Multiplication/Reduction Using SSE

What patterns are you seeing in the speed-ups:

The speed-up is surprisingly consistent across array sizes ranging from 1,024 to 8,388,608. The average speed-up is 2.81, and there is only a small deviance at array size 262,144, with a speed-up of 2.79.

Are they consistent across a variety of array sizes:

The speed-up is consistent across a variety of array sizes possibly because I used a for-loop with 20 NUMTRIES and then used the max performance for each array size, similar to the strategy used in Project #1. I ran multiple tests, however, I did not find a strong pattern relating to the array size and speed-up.

Why, or why not, do you think:

In the lecture for SIMD Vectorization, it was mentioned that as the data size increases, there is often a temporal coherence issue which causes a decrease in speed-up. It could be possible that my data size was not large enough to see a clear pattern in the speed-up.