Project #4: Vectorized Array Multiplication and Multiplication/Reduction using SSE

Sunil Jain

jainsu@oregonstate.edu

CS 475

Professor Bailey

May 14, 2024

1. What machine you ran this on

I ran this on a Windows 11 laptop where I SSH’ed into flip server 1.

1. Show the 2 tables of performances for each array size and the corresponding SPEEDUPs.

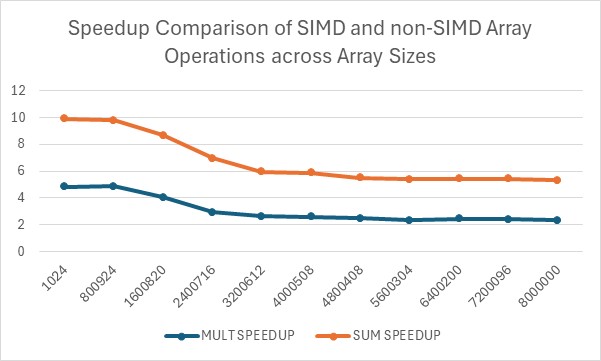
Multiply Operator vs Array Size Table



Addition Operator vs Array Size Table



1. Show the graphs (or graph) of SIMD/non-SIMD SPEEDUP versus array size (either one graph with two curves, or two graphs each with one curve)



1. What patterns are you seeing in the SPEEDUPs?

For both array multiplication and summation, the SPEEDUP decreases as array size increases. The SPEEDUP values are consistently above 1, meaning that the SIMD implementation outperforms the non-SIMD implementation.

I believe that the MULT SPEEDUP is approaching somewhere between 2-3 while the SUM SPEEDUP is approaching 3.

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

The SPEEDUP values are consistent across different array sizes, both multiplication and summation shows a decrease in SPEEDUP as array size increases.

The rate of decrease in SPEEDUP as array size increases varies.

1. Why or why not, do you think?

The variations in the rate of decrease in SPEEDUP across array sizes, can be attributed to external factors like flip server uptime and others.

As array size increases, the efficiency of the SIMD operations are significantly reduced due to memory limitations, cache limitations, and overhead.

Overall, while SIMD increases performance, the variation in SPEEDUP across array sizes highlight the why optimization is important.