An-Najah National University
Computer Engineering Department
Parallel Processing - 10636523
SIMD Vectorization

Student name : Raghad Shaar
Registration number : 12217003
Instructer: dr.Samer Arandi

	Vector-Vector Multiplication		
Input-size	Scalar	Vector	improvement(%)
128	0.000000724	0.000000358	2.02x
256	0.000001404	0.000000685	2.05x
512	0.00000231	0.00000114	2.03x
	Matrix-Vector Multiplication		
Input-size	Scalar	Vector	improvement(%)
128	0.000083	0.000042	1.98x
256	0.000294	0.000146	2.01x
512	0.001050	0.000486	2.16x
	Matrix-Matrix Multiplication		
Input-size	Scalar	Vector	improvement(%)
128	0.008871	0.004232	2.10x
256	0.074260	0.027565	2.69x
512	0.682	0.237	2.9x
Improvement(%) = Scalar/Vector			
When compiling use: gcc -O0			

Performance Analysis of SIMD (SSE) vs Scalar Operations

■ Matrix-Matrix Multiplication

Matrix-matrix multiplication demonstrated the most significant performance gains with SIMD optimization. At the largest input size of 512, the vectorized implementation achieved a remarkable 2.9× speedup over the scalar version. This substantial improvement stems from the operation's computational intensity, featuring three nested loops arithmetic operations that perfectly match SIMD's parallel processing capabilities. The results clearly show that matrix multiplication represents an ideal case for SIMD acceleration, with benefits increasing proportionally to problem size due to improved instruction-level parallelism and cache utilization.

■ Matrix-Vector Multiplication

Matrix-vector multiplication exhibited stable performance improvements averaging ~2× speedup across sizes (1.98× – 2.16×). This operation benefits more substantially from SIMD than simple dot products due to its inherent row-wise parallelism, where multiple elements can be processed simultaneously. While the gains aren't as dramatic as with full matrix multiplication, the consistent 2× acceleration demonstrates SIMD's effectiveness for this medium-complexity operation. Such improvements are particularly valuable as matrix-vector products form the backbone of many numerical algorithms and machine learning applications.

■ Vector-Vector Multiplication (Dot Product)

The dot product operation showed the most modest improvements, maintaining approximately 2× speedup regardless of input size. Being fundamentally memory-bound, its performance is constrained by memory bandwidth limitations rather than raw compute power. While SIMD does provide meaningful acceleration by processing multiple elements concurrently, the operation's simple structure and memory access patterns prevent it from achieving the same dramatic gains seen in more computationally intensive operations. Nevertheless, even these moderate improvements can yield significant benefits when accumulated across the millions of dot products performed in typical scientific computing workloads.

System Specifications

Component	Details
CPU	Intel Core i7-8550U (8th Gen) - 4 Cores / 8 Threads
Cache Memory	L1: 128KB (Instruction) + 128KB (Data) L2: 1MB L3: 8MB (Shared)
RAM	16GB DDR4 (2400 MHz)
OS	Windows 11 Home (64-bit)
Software Support	WSL (Windows Subsystem for Linux)