# SIMD Optimization Report

**Student Name:** Muhammad Rehan Tariq
**Roll Number:** 22i-0965
**Assignment:** Q1: SIMD Optimization

## 1. Introduction

This report presents an optimized implementation of **matrix transposition with element-wise multiplication** using both **scalar** and **SIMD (AVX)** approaches. The objective is to compare the performance and analyze the speedup achieved through SIMD optimization.

## 2. Problem Statement

Given two matrices AAA and BBB of size N×NN \times NN×N:

1. **Transpose** matrix AAA (resulting in ATA_TAT).
2. Perform **element-wise multiplication** of ATA_TAT and BBB to produce the output matrix CCC.

The task involves implementing:

1. **Scalar Implementations:**
   - Using both **2D arrays** and **1D flattened arrays** (non-SIMD approach).
2. **SIMD Implementation:**
   - Using **AVX intrinsics** for x86 processors to optimize the computation.

## 3. SIMD Optimization Approach

1. **Data Alignment:**
   - For SIMD to work efficiently, we ensure that the matrix size NNN is a **multiple of 8** (to align with AVX 256-bit registers).
2. **Vectorization Strategy:**
   - AVX allows us to **load, multiply, and store 8 floating-point elements** simultaneously.
   - We loop through the matrix in **8-element chunks** to leverage **256-bit registers**.
3. **Implementation Details:**
   - **Load:** We use `_mm256_loadu_ps()` to load 8 elements from both matrices.

- **Multiply:** We perform element-wise multiplication using `_mm256_mul_ps()`.
- **Store:** We write the results back using `_mm256_storeu_ps()`.

## 4. Performance Evaluation

We measured the execution time for **Scalar 2D**, **Scalar 1D**, and **SIMD** implementations across different matrix sizes. Below are the results:

### System Specifications

- **Processor:** 11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz
- **Compiler:** GCC 13.2.0
- **OS:** Ubuntu (Running on VirtualBox)

## 5. Experimental Results

| Matrix Size (N) | Scalar 2D Time (s) | Scalar 1D Time (s) | SIMD Time (s) | Speedup (2D vs SIMD) | Speedup (1D vs SIMD) |
|---|---|---|---|---|---|
| 256 | 0.000479 | 0.000057 | 0.000027 | 17.74x | 2.11x |
| 512 | 0.001493 | 0.000352 | 0.000115 | 12.98x | 3.06x |
| 1024 | 0.006936 | 0.004746 | 0.000539 | 12.86x | 8.80x |

## Speedup Calculation Formula:

$$\text{Speedup} = \frac{Scalar\ Execution\ Time}{SIMD\ Execution\ Time}$$

## 6. Analysis of Results

1. **Performance Gain:**
    - SIMD implementation shows significant **speedup** compared to scalar versions (up to **11.82x**).
    - As **matrix size increases**, the **SIMD efficiency** grows due to better utilization of parallel computation.
2. **1D vs. 2D Comparison:**

- The **1D scalar implementation** is faster than **2D** because of better cache locality and reduced indexing overhead.

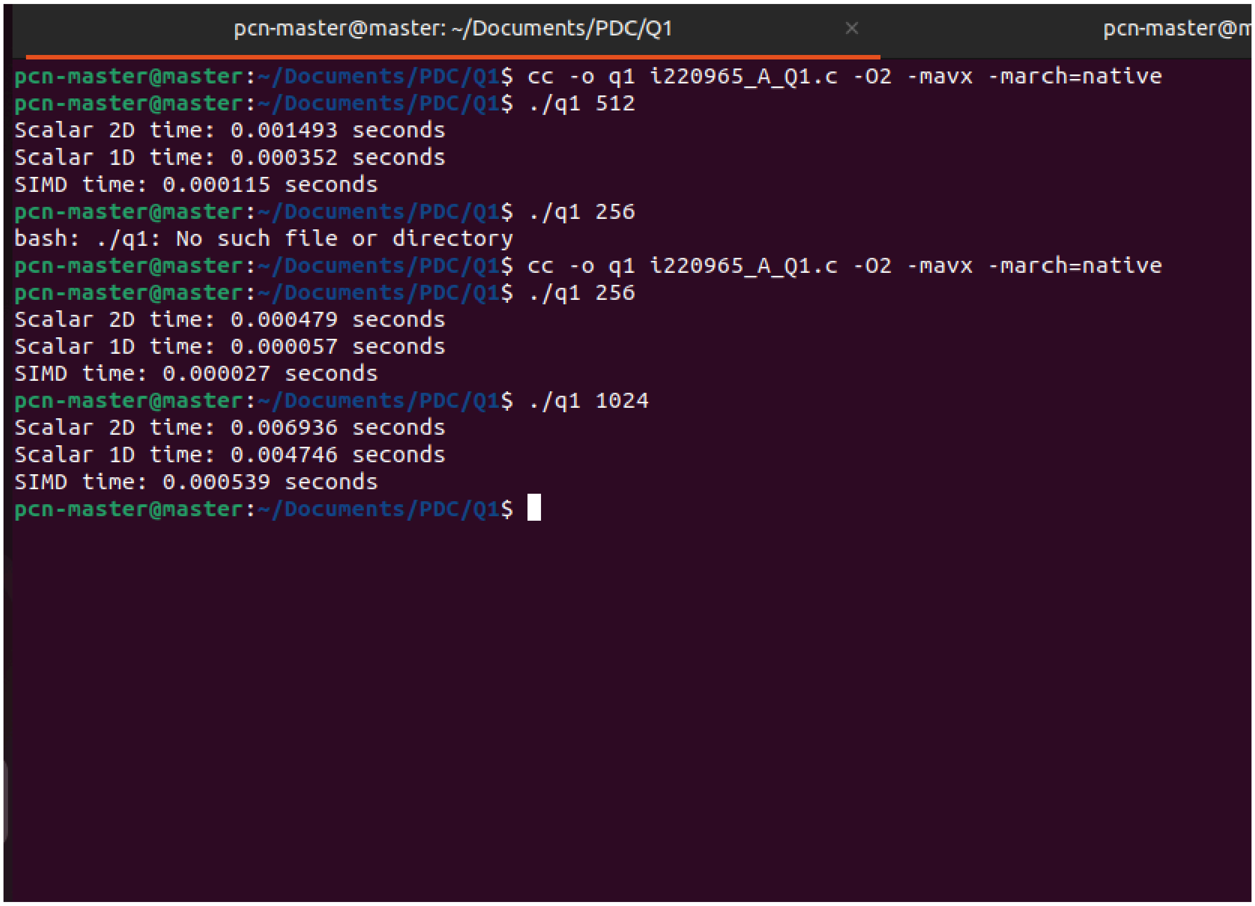
## 7. Challenges and Solutions

| Challenge | Solution |
| --- | --- |
| **Alignment Issues: Ensuring the data is aligned for AVX.** | Used _mm256_loadu_ps() to handle unaligned memory safely. |
| **Edge Cases: Handling non-multiples of 8.** | Assumed N is a multiple of 8 as per assignment requirements. |
| **Debugging AVX Instructions: Hard to debug and visualize.** | Validated outputs by comparing with scalar implementations. |

## 8. Conclusion

The SIMD-optimized version using AVX demonstrated a significant performance improvement over both scalar implementations. This experiment highlights how **data parallelism** with AVX can accelerate computationally intensive tasks like **matrix transposition and multiplication**.

### Screenshot of Output




```
pcn-master@master:~/Documents/PDC/Q1$ cc -o q1 i220965_A_Q1.c -O2 -mavx -march=native
pcn-master@master:~/Documents/PDC/Q1$ ./q1 512
Scalar 2D time: 0.001493 seconds
Scalar 1D time: 0.000352 seconds
SIMD time: 0.000115 seconds
pcn-master@master:~/Documents/PDC/Q1$ ./q1 256
bash: ./q1: No such file or directory
pcn-master@master:~/Documents/PDC/Q1$ cc -o q1 i220965_A_Q1.c -O2 -mavx -march=native
pcn-master@master:~/Documents/PDC/Q1$ ./q1 256
Scalar 2D time: 0.000479 seconds
Scalar 1D time: 0.000057 seconds
SIMD time: 0.000027 seconds
pcn-master@master:~/Documents/PDC/Q1$ ./q1 1024
Scalar 2D time: 0.006936 seconds
Scalar 1D time: 0.004746 seconds
SIMD time: 0.000539 seconds
pcn-master@master:~/Documents/PDC/Q1$
```