Team ROR

Rodaina Mohamed 34-7750 T10

Omar Doma 34-12601 T10

Reem Eslam 34-3184 T14

MileStone 2 Report

a)

```
__m128 *lvfA, *lvfB;
float *lafResult;
for (int i=0; i < MAX_DIM; i++)
{
    for (int j = 0; j < MAX_DIM; j++)
    {
        for (int k = 0; k < MAX_DIM; k += 4)
        {
            lvfA = (__m128 *)&a[i];
            lvfB = (__m128 *)&b[j];
            __m128 lvfA_mul_B = _mm_mul_ps(*lvfA, *lvfB);
            lafResult = (float*)&lvfA_mul_B;
          d[i][j] += lafResult[0] + lafResult[1] + lafResult[2] + lafResult[3];
        }
        lvfA++;
        lvfB++;
    }
}
```

b) Performance of code using vector parallelization is much better than loop performance in terms of execution time, as our code is executing at a much shorter time than the loop execution time.

c) As matrix dimension increases, our code performance is stable and still executing in less time than the loop, however loop code performance degrades and execution time gets longer.

d) Intrinsic functions we used:

__m128 _mm_mul_ps (__m128 a, __m128 b)

To multiply packed single-precision (32-bit) floating-point elements in a and b, and store the results in dst.

e) Vector parallelization can be used in Many applications such as video processing (GAPP), video Games, Cell Processors and many other multimedia applications.