Consider the following code segment, with the values MAX = 2^{20} and WSIZE = 5 and the following data:

- Read from x takes 10 cycles (accounts for lw add)
- Read from w takes 10 cycles (accounts for lw add)
- Write to a register takes 1 cycle.
- Multiply takes 10 cycles (accounts for writing to the register)
- Each addition takes 5 cycles (accounts for writing to the register)
- Write to y takes 10 cycles
- Each comparison takes 5 cycles
- L1 cache hits require 5 cycles
- Cache miss penalty is 10 cycles

```
for( i = 0; i < MAX; ++i ){
        t = 0;
        for( j = 0; j < WSIZE; ++j ){
                t += x[i+j]*w[j];
                y[i] = t;
        }
}
```

a) Without any modifications, how many cycles does the code segment take?

for i: $i=0 \Rightarrow$ Once, $i < MAX \Rightarrow 5\times 2^{20}$, $++i \Rightarrow 5\times 2^{20}$, $t=0 \Rightarrow 1\times 2^{20} = \boxed{11\times 2^{20}+1 \text{ cycles}}$

for j: $j=0 \Rightarrow$ Once $\Rightarrow 2^{20}$, $j < WSIZE \Rightarrow \underline{5\times} 5\times 2^{20}$, $++j \Rightarrow \underline{5\times} 5\times 2^{20} \Rightarrow \boxed{51\times 2^{20} \text{ cycles}}$

$x(i+j) \Rightarrow i+j \Rightarrow \underline{5\times} 5\times 2^{20}$, $w[i+j) \Rightarrow \underline{10\times} 5\times 2^{20}$, $w[i] = \underline{10\times} 5\times 2^{20}$,

Multiply : $\underline{10\times} 5\times 2^{20}$, Add: $\underline{5\times} 5\times 10^{20}$

y: $\underline{10\times 5\times} 2^{20}$

$(11+51+25+50+50+50+25+50)\times 2^{20}+1$

$\boxed{312\times 2^{20}+1 \text{ cycles}}$

Consider the following code segment, with the values MAX = 2^{20} and WSIZE = 5 and the following data:

- Read from x takes 10 cycles (accounts for lw add)
- Read from w takes 10 cycles (accounts for lw add)
- Write to a register takes 1 cycle.
- Multiply takes 10 cycles (accounts for writing to the register)
- Each addition takes 5 cycles (accounts for writing to the register)
- Write to y takes 10 cycles
- Each comparison takes 5 cycles
- L1 cache hits require 5 cycles
- Cache miss penalty is 10 cycles

```
for( i = 0; i < MAX; ++i ){
        t = 0;
        for( j = 0; j < WSIZE; ++j ){
                t += x[i+j]*w[j];
                y[i] = t;
        }
}
```

Load w(0) from w(4) into registers

$\boxed{50 \text{ cycles}}$ total

b) Discuss the opportunities for reduction of cycle times and determine the improvement in performance when implemented in SIMD with 8 processors. → Assume strong scaling

for i → Still $\boxed{10 \times 2^{20} + 1}$ cycles

$t = 0 \rightarrow \boxed{1 \times 2^{20}}$

Load x(i) → x(i+4) into cache → Reduces load from 10 to 5

$t = x[i] \times w0 \rightarrow (5+10) \times 2^{20} = \boxed{15 \times 2^{20} \text{ cycle}}$

Other 4: $t \mathrel{+}= x[i+\alpha] \cdot w\alpha \Rightarrow (5+5+10+5) \times 2^{20} = 25 \times 2^{20}$

i+α   load x   multiply   + +=

$4 \times 25 \times 2^{20} = \boxed{100 \times 2^{20} \text{ cycles}}$

Cache miss every cycle: 10×2^{20}

$y(i) = t \Rightarrow \boxed{10 \times 2^{20}}$ cycles

Sum: $(10 + 1 + 15 + 100 + 10 + 10) \times 2^{20} + 50 + 1 \Rightarrow \boxed{146 \times 2^{20} + 51}$

Non-parallel improvement

$\frac{312}{146} \approx \boxed{2.14 \times}$

Parallel w/ Strong Scaling

$\frac{312}{(146/8)} = \boxed{17.09 \times}$