

UNIVERSITY OF CAPE TOWN

EEE4120F - High Performance Digital Embedded System

Quiz 2 - Pthreads and Performance Measurement

14th March 2019

PROBLEM

You are part of an engineering team that is developing an assistive technology for people who have hearing challenges. The system is aimed at helping deaf and hard-hearing people communicate effectively at work-places, school and at home by providing real-time speech to text conversion. The system is being designed to perform robustly under high noise environments and therefore one of the project tasks include design of a high performance finite impulse response filter to denoise the input speech. Given a set of N real filter coefficients, $h = \{h_0, h_1, \dots, h_{N-1}\}$, which specifies the desired behavior of the filter, and real input speech samples buffer of size $Z > N$, $x = \{x_0, x_1, \dots, x_{Z-1}\}$, the filter calculates each filtered output sample $y(t)$ at time $t = \{t_0, t_1, \dots, t_T\}$, by performing a **sum of products** according to the following formula, whose serial C/C++ implementation is also given:

$$y(t) = \sum_{k=0}^{N-1} h(k) * x(t - k)$$

```
void serial_fir(float *h, int N, float *x, int Z, float *y, int T){
    float sop;
    for(int t=0;t<T;t++){//for each output sample
        sop=0;
        for(int k=0;k<N;k++){//consider all filter coefficients
            sop+=(t>=N-1)?h[k]*x[t-k]:0;//perform sum of prod when t>=N-1
        }
        y[t]=sop;//store output sample at time t
    }
}
```

1. What partitioning scheme would you use to accelerate this problem? Task-level, data-level or both? Explain briefly. **[3]**
2. Write a code snippet of a Pthreads kernel, and necessary supporting global and/or local variables and synchronization mechanisms, that you would use to support the partitioning scheme you selected above. **[15]**
3. Upon benchmarking, you discover that, for $Z=1000$, the algorithm runs 3 sec on your platform using compiler A. A new compiler B is released that requires only 0.8 as many instructions as the old compiler. Unfortunately, it increases the CPI by 1.1. How fast do you expect the application to run using this new compiler? **[5]**