

Konstantin Zelmanovich, Claudio Del Valle A

Dr. Nagarajan Kandasamy

ECEC 413 – Introduction to Parallel Computer Architecture

Assignment 3

May 6 2019

### Part 1 - OpenMP: Numerical Integration

In the below code, a pragma was used during the initialization part, allowing for OpenMP to parallelize this functionality. The results of speedup are provided below.

```
double
compute_using_omp (float a, float b, int n, float h, int num_threads)
{
    double integral = 0.0;
    int k;

    // Initialization of the integral from a single thread
    integral = (f(a) + f(b))/2.0;

    // Setting up number of threads
    omp_set_num_threads(num_threads);

    // Parallelization part
    #pragma omp parallel default(none) shared(a, b, h, n) private(k) reduction(+:
integral)
    {
        for (k = 0; k < n-1; k++)
        {
            integral = integral + f(a + k * h);
        }
        // printf("With %d trapezoids, the estimate for the integral between [%f,
%f] is %f \n", n, a, b, integral);
        integral = integral * h;
    }
    return integral;
}
```

**Table 1:** Timing results for 100000000 elements

# of Threads	# of Elements	Serial Time (s)	Parallel Time (s)	Speedup
2	100000000	4.05	2.15	1.89
4	100000000	4.23	1.09	3.88
8	100000000	4.24	0.6	7.08
16	100000000	4.23	0.34	12.29

**Figure 1:** Timing results for 100000000 elements

