

# High Performance Computing

## Assignment 05

**Start:** Tuesday, *February 24th*, 2009

**End:** Thursday, *March 3<sup>rd</sup>*, 2009

### **PART I: (25 Pts)**

A Digital Signal Processor (DSP) is attached to a single processor node and capable of operating at 20 times the performance of the host processor performing an FFT computation. A particular run of the user application makes 100 calls to the DSP (the inner loop iterates 100 times). Each iteration of the FFT algorithm found in the inner loop of a user application takes second to execute on the host processor. For each call to the DSP, a start up overhead of 40 milliseconds is required and another 10 milliseconds is needed to offload the results produced by the DSP. The host equipped with the DSP accelerator executes the application 2.5 times faster than the host processor alone. The remainder of the application is sequential and independent of the input data set size. What is the duration (in seconds) of the sequential part of the application? Show all work.

### **PART II: (15 Pts)**

Describe the memory hierarchy of a processor and discuss the roles and relationships of the different levels.

### **PART III: (60 Pts)**

Write a C program using both MPI and OpenMP to compute in parallel the Simpson's rule, for the function  $f(x) = \sin(x)$  over the interval  $a = \pi/8$  to  $b = 5\pi/8$

Simpson's Rule:

$$\int_a^b f(x)dx \approx \frac{h}{3} [f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + 2f(x_4) + \dots + 4f(x_{n-1}) + f(x_n)]$$

With:

$$h = \frac{b - a}{n}$$

And

$$x_i = a + ih \text{ for } i = 0, 1, 2, \dots, n - 1, n$$

$n$  is a user input argument.

You are to submit:

- Source codes of the MPI and OpenMP programs
- The .pbs submit files of the MPI and OpenMP programs
- A readme file that briefly describe the structure of the program, any special instruction on compiling and running the program, and the method that you used to determine that your program is correct.

- For 4 and 8 processors, run the following values of n: 100'000, 500'000, 1'000'000 (total of 6 runs) and return the outputs.