EECE5640
High Performance Computing
Homework 3
*Submit your work on Canvas in a single zip file.

1. (40) In this problem, you will utilize the IEEE 754 format and evaluate the performance implications of using floats versus doubles in a computation.

   a.) Compute f(x) = sin (x) using a Taylor series expansion. To refresh your memory:

   sin(x) = $\sum_0^{\infty} \frac{(-1)^n}{(2n+1)!}\, x^{\,2n+1}$

   sin(x) = $x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} \ldots.$

   You select the number of terms you want to compute (but at least 10 terms). Compute sin(x) for 4 different values, though be careful not to use too large a value. Generate two versions of your code, first defining x and sin(x) to use floats (SP), and second, defining them as doubles (DP). Discuss any differences you find in your results for f(x). You should provide an in-depth discussion on the results you get and the reasons for any differences.

   b.) Explore the benefits of compiling on the Discovery cluster with floating point vector extensions (e.g., AVX). Use the single-precision code from part (a). First run on a node on Discovery that does not support AVX-512. Then run on a node that supports AVX-512 and report on the performance benefits. Additional information is provided on AVX support on Discovery.

   c.) Continuing with part (b), generate an assembly listing (using the -S flag) and identify 2 different AVX instructions that the compiler generated, explaining their operation.

   d.) Provide both IEEE 754 single and double precision representations for the following numbers: 2.1, 6300, and -1.044.

2. (30) In this problem, you will modify the matmul.c program provided, optimizing the execution of the matrix multiplication with first a dense matrix, and second with a sparse matrix. You are welcome to use pthreads, OpenMP or any of the optimizations that were presented in class to accelerate this code. There will be prizes awarded for the fastest dense and the fastest sparse implementations.

3. (30) In this problem, you will utilize the OpenBLAS library available on Discovery. To use OpenBLAS, you will need to issue load openblas/0.3.6. Using the malmul.c program, replace the math with a call to appropriate gemm library function. Compare the speed of your solution for problem 2 with the gemm method you used.

(15 points for MS and 25 points for Undergraduate/Plus-One) (Extra quiz credit for everyone) Find a published paper from an ACM or IEEE conference that discusses a novel sparse matrix format that was not covered in class. Discuss why the proposed format is superior to the CSR or CSC format. Make sure cite your sources.

* Written answers to the questions should be included in your homework 3 write-up in pdf format. You should include your C/C++ programs and the README file in the zip file submitted.