OpenMP Problems

7th December 2019

Problem 1

(a) Modify the optimized() function in file mat-vector.cpp to implement an OpenMP version of the reference kernel.

(b) What is the maximum speedup you can expect from your code?

Problem 2

Implement a parallel version of Fibonacci sum using OpenMP tasks. Try to optimize your parallel code for maximum speedup.

Problem 3

Assume an array of type int and the size of the array is 2^{24}. Implement computing the sum of the elements of an array using the following strategies:

(a) OpenMP parallel for *without* reductions. You are allowed to use synchronization constructs like critical or atomic.

(b) an OpenMP parallel for *using* reductions,

(c) OpenMP tasks where each task will reduce a sub-array of size 1024.

Problem 4

Bubble sort is a naïve way to sort a list of integers. However, it is not very amenable to parallelization since there are dependences across iterations. An alternative is *odd-even* transposition sort that provides more opportunities for parallelization. Parallelize odd-even transposition sort with OpenMP.