

Threaded Programming

Coursework Part 1

Overview

- Parallelise a simple piece of code using OpenMP
- Worth 10% of the total marks for the course
- You may choose to work with either the C or the Fortran version.
- Deadline is 16:00, Fri 16th October
- Standard late penalties will apply
- Details of submission process to follow

Technical details

- Add OpenMP directives to parallelise all the computation between the two calls to the timer routine `omp_get_wtime()`.
- You are *not* expected to otherwise optimise the code, or modify it except for the addition of OpenMP
- Your solution will be marked for correctness, by compiling it with the Intel 19 compilers, as in the supplied Makefile and running it on up to 36 threads on Cirrus.
 - The diff value printed should be identical to that of the sequential code for all thread numbers.

- You will be given credit for keeping the changes made to the original C or Fortran code to a minimum.
 - you are *not* intended to spend time trying to improve the performance beyond that of a straightforward solution.
- Your code will *not* be marked directly for performance
 - you will be given additional credit for minimising the number of implicit barriers that are executed
 - you may be penalised if the performance is very poor (i.e. worse than the sequential code).