Introduction

This is your first assessed exercise contributing 50% of your overall mark. This is an individual exercise and you will have approximately one week to complete it. The total mark for the exercise if 100. The marks assigned to each part of the exercise are put in brackets.

Part 1 [40 marks total]

The following declarations and definitions occur in all tasks before the code snippets provided:

```
int i;
double A[N] = { ... }, B[N] = { ... }, C[N], D[N];
const double c = ...;
const double x = ...;
double y;
```

Task 1 [10 marks]

Insert missing OpenMP directives to parallelise this loop:

```
for (i = 0; i < N; i++)
{
   y = sqrt(A[i]);
    D[i] = y + A[i] / (x * x);
}
```

Task 2 [10 marks]

Insert missing OpenMP directives to make both loops run in parallel:

```
{
   for (i = ; i < N; i += )
      {
         D[i] = x * A[i] + x * B[i];
      }

   #pragma omp for
   for (i = 0; i < N; i++)
   {
       C[i] = c * D[i];
   }
} // end omp parallel
```

Task 3 [10 marks]

Can you parallelise this loop – if yes how, if not why?

```
#pragma omp parallel for
for (int i = 1; i < N; i++)
{
 A[i] = B[i] - A[i - 1];
}
```

The mark awarded will be reduced by 10% for each delay in submission of a day or partial day.

Task 4 [10 marks]

The code snippet below implements a Matrix times Vector (MxV) operation, where a is a vector of R^m, B is Matrix of R^{mxn} and c is a Vector of R^n: a = B * c.

```
01 void mxv_row(int m, int n, double *A, double *B, double *C)
02 {
03    int i, j;
04
05    for (i=0; i<m; i++)
06    {
07       A[i] = 0.0;
08       for (j=0; j<n; j++)
09          A[i] += B[i*n+j]*C[j];
10    }
11 }
```

Parallelise the for loop in line 05 by providing the appropriate line in OpenMP. Which variables must be private, and which shared and why?

Part 2 [60 marks total]

- Implement one searching or sorting algorithm in non-parallel and parallel (using OpenMP) execution formats – try to make your parallel version to execute as fast as possible **[20 marks]**
- Compare the speedup gained (compared to the serial version) when the parallel work is spread across different number of threads (from the least possible to the max possible threads) **[20 marks]**
- What is the maximum speedup you can expect from your parallel program? **[20 marks]**

Submission guidelines and viva

- Submission
  - The submission deadline is 10am on Monday 29 October
  - Your submission will be on Blackboard
  - Your submission should include the following:
    - A report 2-4 (max) page report (no covers) where
      - For Part 1, you provide answers to the individual tasks
      - For Part 2, you present your results (don't include full code, just a few snippets if necessary) and make comments about your implementation decisions and results
    - 1 C file with solutions to the 4 tasks of Part 1
    - 2 C files (1 of the sequential version and 1 of the OpenMP version of your algorithms) for Part 2
    - **Important:** You should zip all the above to a single archive (e.g. zip or .rar) - this is what you will submit on Blackboard
- There will also be a short viva on Tuesday (30th of October)

Dr Ioannis Kyriakopoulos

22 October 2018