Homework Assignment #5

Due: Sunday, November 13, 2022

1. **Try the codes in directories pi and hybrid.**
2. **Implement the matrix multiplication program with OpenMP**

you are to add OpenMP constructs to the sequential program for matrix multiplication given

here:

#include <omp.h>

#include <stdio.h>

#include <stdlib.h>

#define M 500

#define N 500

#define Max\_threads 8

int main(int argc, char \*argv) {

omp\_set\_num\_threads(Max\_threads);//set number of threads here

int i, j, k;

double sum;

double start, end; // used for timing

double \*\*A, \*\*B, \*\*C;

A = malloc(M\*sizeof(double \*));

B = malloc(M\*sizeof(double \*));

C = malloc(M\*sizeof(double \*));

for (i = 0; i < M; i++) {

A[i] = malloc(N\*sizeof(double));

B[i] = malloc(N\*sizeof(double));

C[i] = malloc(N\*sizeof(double));

}

//initialize matrix A, B, C

for (i = 0; i < M; i++) {

for (j = 0; j < N; j++) {

A[i][j] = j\*1;

B[i][j] = i\*j+2;

C[i][j] = j-i\*2;

}

}

start = omp\_get\_wtime(); //start time measurement

for (i = 0; i < M; i++) {

for (j = 0; j < N; j++) {

sum = 0;

for (k=0; k < M; k++) {

sum += A[i][k]\*B[k][j];

}

C[i][j] = sum;

}

}

end = omp\_get\_wtime(); //end time measurement

printf("Time of computation: %f\n", end-start);

}

You are to parallelize this algorithm in three different ways:

1. Add the necessary pragma to parallelize the outer for loop;
2. Remove the pragma for the outer for loop and create a pragma for the middle for loop;
3. Add the necessary pragma’s to parallelize both the outer and middle for loops.
4. Add an assert function that compares the parallel computing result with serial computing result in order to make sure that you get correct result.

You need to collect timing data given one thread, four threads, eight threads, and 16 threads and two matrix sizes, 50x50 and 500x500. You will find that when you run the same program several times, the timing values can vary significantly. Therefore, for each set of conditions, collect ten data values and average them. You should write a report and analyze the results.

Here are the conditions you should collect data for:

1. No parallelization at all (that is, the given program)
2. Parallelizing the loops as above with 1, 4, 8, and 16 threads using matrix sizes 50x50 and 500x500