ME-489 HOMEWORK 3 REPORT

Alperen Çalışkan
Ali Taha Akpınar

December 6, 2023

Abstract

This report is a documentation of the explanation of ME489 Homework 3 code's parallelization part. It explains the implementation of parallelization of the given code. It starts with the parallellization of RhsQ function and continues with the infinity norm of the right hand side solution vector.

1 Introduction

The problem in the homework is about an explicit in-time solver which uses finite difference method. The programming language of interest is C. The core part of the code is corrected and completed by the instructor. The instructor has created a code which takes input data, creates mesh nodes accordingly, creates time step, applies fourth order Runge Kutta method, Integrates over time, creates initial conditions for the nodes (given by a known equation), creates periodic connectivity between each element by using a mesh function and uses a RhsQ function to obtain field data for each node in context over each time step. In our homework, the RhsQ function is paralleled by the given number of threads from the person executing the program which should be the third argument(argv[2]). Creates that amount of threads and by implementing it on the for loop using a pragma omp statement. Another change in the code is to create an infinity norm for the field variable. It is asked to create a different function for this purpose and use multiple methods such as critical and reduction.

2 RhsQ function

The right hand side of Q function is already implemented a pragma statement is created after the function starts.

```
1 #pragma omp parallel for default(none) shared(solver,msh,tstep,stage)
```

The parallel for statement automatically divides the for loop into the defined thread size and iterates over them individually. By using this statement the same field variable results are achieved for each time step compared with the nonparallelized code given by the instructor.

3 Infinity Norm function

The infinity norm gives the highest value of field variable which, in our case is Q. In the main function to the time integration part just like the solverPlot an InfNorm function is put for getting the maximum values in every time step.
After that the function itself is written. The function takes msh and double Q as inputs and prints the infinity norm at every time step. Because four different cases are used, the infinity norm is printed out 4 times. For every case time is recorded thus each time a pragma statement of critical, reduction, only parallel for or nothing at all is used the time is printed.

4 Time records

The differences in time, using different methods are discussed here.

4.1 When 2 threads are used

When 2 threads are used, the following results are attained for time steps 0,10,90,100 respectively. In the following context max1 indicates where no parallelization is used, max2 indicates only parallel for statement is used, max3 indicates to critical method and max4 is when reduction is used. Time step is 0.

```
threadcount: 2
Max1:0.75138440
The time with no parallelization:0.000081
Max2:0.75138440
The time with only parallel for:0.000073
Max3:0.75138440
The time with critical:0.001017
Max4:0.75138440
The time with reduction0.000039
```

Time step is 10.

```
Max1:2.29035049
The time with no parallelization:0.000106
Max2:2.29035049
The time with only parallel for:0.000099
Max3:2.29035049
The time with critical:0.001037
Max4:2.29035049
The time with reduction0.000042
```

Time step is 90.

```
Max1:91.19786744
The time with no parallelization:0.000158
Max2:91.19786744
The time with only parallel for:0.000146
Max3:91.19786744
The time with critical:0.001789
Max4:91.19786744
The time with reduction:0.000063
```

Time step is 100.

```
Max1:101.71180252
The time with no parallelization:0.000132
Max2:101.71180252
The time with only parallel for:0.000235
Max3:101.71180252
The time with critical:0.000756
Max4:101.71180252
```

```
The time with reduction:0.000063
parallel for shared took: 363.022705
```

It is clear that as time step goes forward the infinity norm tends to increase more each time. It is also visible that reduction is the fastest option among all followed by no parallelization, parallel for and finally critical with highest time record each time step.

4.2 When 4 threads are used

When 4 threads are used, the following results are attained for time steps 0,10,90,100 respectively. Time step is 0.

```
threadcount: 4
Max1:0.75138440
The time with no parallelization:0.000078
Max2:0.75138440
The time with only parallel for:0.000330
Max3:0.75138440
The time with critical:0.002000
Max4:0.75138440
The time with reduction:0.000044
```

Time step is 10.

```
Max1:2.02227407
The time with no parallelization:0.000102
Max2:2.02227407
The time with only parallel for:0.000157
Max3:2.02227407
The time with critical:0.002413
Max4:2.02227407
The time with reduction:0.000041
```

Time step is 90.

```
Max1:91.19786743
The time with no parallelization:0.000121
Max2:91.19786743
The time with only parallel for:0.000142
Max3:91.19786743
The time with critical:0.002725
Max4:91.19786743
The time with reduction:0.000046
```

Time step is 100.

```
Max1:101.71180250
The time with no parallelization:0.000104
Max2:101.71180250
The time with only parallel for:0.000209
Max3:101.71180250
The time with critical:0.002565
```

```
Max4:101.71180250
The time with reduction:0.000059
parallel for shared took: 299.304321
```

It is clear that as time step goes forward the infinity norm tends to increase more each time. It is observed that reduction is the fastest approach followed by no parallelization, only parallel and critical.

4.3 When 16 threads are used

When 16 threads are used, the following results are attained for time steps 0,10,90,100 respectively. Time step is 0.

```
threadcount: 16
Max1:0.75138440
The time with no parallelization:0.000079
Max2:0.75138440
The time with only parallel for:0.000268
Max3:0.75138440
The time with critical:0.001937
Max4:0.75138440
The time with reduction0.000282
```

Time step is 10.

```
Max1:2.29035017
The time with no parallelization:0.000092
Max2:2.29035017
The time with only parallel for:0.000169
Max3:2.29035017
The time with critical:0.002058
Max4:2.29035017
The time with reduction0.000112
```

Time step is 90.

```
Max1:91.19784677
The time with no parallelization:0.000139
Max2:91.19784677
The time with only parallel for:0.000167
Max3:91.19784677
The time with critical:0.002803
Max4:91.19784677
The time with reduction0.000132
```

Time step is 100.

```
Max1:101.71179576
The time with no parallelization:0.000106
Max2:101.71179576
The time with only parallel for:0.000208
Max3:101.71179576
The time with critical:0.002714
```

```
Max4:101.71179576
The time with reduction0.000262
parallel for shared took: 386.013794
```

The time required for each approach has increased and the reduction approach has become slower than the no parallel approach. The overall time of all operations has increased by almost 1.25 times as well.

4.4 When 64 threads are used

When 64 threads are used, the following results are attained for time steps 0,10,90,100 respectively. Time step is 0.

```
threadcount: 64
Max1:0.75138440
The time with no parallelization:0.000079
Max2:0.75138440
The time with only parallel for:0.000442
Max3:0.75138440
The time with critical:0.002175
Max4:0.75138440
The time with reduction0.000335
```

Time step is 10.

```
Max1:2.29032413
The time with no parallelization:0.000088
Max2:2.29032413
The time with only parallel for:0.000349
Max3:2.29032413
The time with critical:0.002258
Max4:2.29032413
The time with reduction0.000369
```

Time step is 90.

```
Max1:91.19779377
The time with no parallelization:0.000119
Max2:91.19779377
The time with only parallel for:0.000469
Max3:91.19779377
The time with critical:0.003049
Max4:91.19779377
The time with reduction0.000487

```

Time step is 100.

```
Max1:101.71181159
The time with no parallelization:0.000139
Max2:101.71181159
The time with only parallel for:0.000503
Max3:101.71181159
```

```
The time with critical:0.003013
Max4:101.71181159
The time with reduction0.000416
parallel for shared took: 344.316162
```

The time record again increased significantly

5 Conclusion

It is observed that time taking to calculate all the Approaches have increased every time thread count increased from 4 threads to 16 threads. This is mainly because the number of processors reserved for the virtualbox does not exceed 4, thus the thread count after that does not only reduce efficiency but decreases it. This may be due to racing condition where threads have to wait for each other to finish updating.