Documentation for exercises

The Documentation of Lab 1 exercises

From: Chaolun Wang
Date: 09/06/2016

This folder includes 5 files/folders:
Documentation_for_exercise.pdf: which is the documentation of the programs
hello.cpp: which is the c++ source code file for lab exercise, the modify was done by Chaolun Wang in 09/06/2016
md.cpp: which is the c++ source code file for lab exercise, the modify was done by Chaolun Wang in 09/06/2016
quad.cpp: which is the c++ source code file for lab exercise, the modify was done by Chaolun Wang in 09/06/2016
jacobi.pdf: which is the report of the assignment, containing the execution time and the discussion of this lab

To compile and run the code:
linux environment with g++ compiler is recommended. Open_MP library needs to be installed to the system. In linux terminal, firstly type:

g++ filename.cpp -fopenmp

to compile the code and link with openmp library, after this command, the a.out should be generated. Then change the file name by the command:

mv a.out filename

After that, the environment thread number is setted by the command:

export OMP_NUM_THREADS=1

The number 1 can be replaced by any preferred thread numbers. And the program is executed by entering:

./filename

Test cases:
The programs were runned three times with the thread number to be 1, 2 and 4. The programs were runned on linux virtual machine installed on personal computer from Chaolun Wang. The virtual machine contain 2 processors and can run maximum 2 threads simultaneously.

Result:

For quad.cpp:
Test case 1, thread number=1:

QUAD:
  C++ version

  Estimate the integral of f(x) from A to B.
  f(x) = 50 / (pi * (2500 * x * x + 1)).
  A = 0
  B = 10
  Exact integral from 0 to 10 is 0.49936338107645674464...

  Estimate = 0.499442
  Error   = 7.90784e-05
  Wallclock time   = 0.00633148

Test case 2, thread number=2:

QUAD:
  C++ version

  Estimate the integral of f(x) from A to B.
  f(x) = 50 / (pi * (2500 * x * x + 1)).
  A = 0
  B = 10
  Exact integral from 0 to 10 is 0.49936338107645674464...

  Estimate = 0.499442
  Error   = 7.90784e-05
  Wallclock time   = 0.00331847

Test case 3, thread number=4:

QUAD:
  C++ version

  Estimate the integral of f(x) from A to B.
  f(x) = 50 / (pi * (2500 * x * x + 1)).
  A = 0
  B = 10
  Exact integral from 0 to 10 is 0.49936338107645674464...

  Estimate = 0.499442
  Error   = 7.90784e-05
  Wallclock time   = 0.00432413

For md.cpp

Test case 1, thread number=1:

MD
C++/OpenMP version

A molecular dynamics program.

NP, the number of particles in the simulation is 500
STEP_NUM, the number of time steps, is 100
DT, the size of each time step, is 0.0001

Initialize positions, velocities, and accelerations.

Compute initial forces and energies.

Initial Total energy = 124434

At each step, we report the potential and kinetic energies.
The sum of these energies should be a constant.
As an accuracy check, we also print the relative error
in the total energy.

Step	Potential Energy	Kinetic Energy	(P+K-E0)/E0 Energy Error
1	124434	0	0
2	124434	1.726e-06	1.38708e-11
3	124434	1.5534e-05	1.38693e-11
4	124434	4.315e-05	1.38718e-11
5	124434	8.4574e-05	1.38698e-11
6	124434	0.000139806	1.38708e-11
7	124434	0.000208846	1.38698e-11
8	124434	0.000291694	1.38672e-11
9	124434	0.00038835	1.38682e-11
10	124434	0.000498814	1.38693e-11
11	124434	0.000623086	1.38684e-11
12	124434	0.000761167	1.38685e-11
13	124434	0.000913055	1.38706e-11
14	124434	0.00107875	1.38653e-11
15	124434	0.00125826	1.38655e-11
16	124434	0.00145157	1.38648e-11
17	124434	0.00165869	1.38678e-11
18	124434	0.00187962	1.38657e-11
19	124434	0.00211436	1.38638e-11
20	124434	0.0023629	1.38602e-11
21	124434	0.00262526	1.38611e-11

22	124434	0.00290142	1.38582e-11
23	124434	0.00319139	1.38573e-11
24	124434	0.00349517	1.38561e-11
25	124434	0.00381276	1.38539e-11
26	124434	0.00414415	1.3852e-11
27	124434	0.00448936	1.38506e-11
28	124434	0.00484837	1.38441e-11
29	124434	0.00522119	1.38392e-11
30	124434	0.00560782	1.38363e-11
31	124434	0.00600826	1.38387e-11
32	124434	0.00642251	1.38312e-11
33	124434	0.00685057	1.38299e-11
34	124434	0.00729243	1.38275e-11
35	124434	0.00774811	1.3819e-11
36	124434	0.00821759	1.38132e-11
37	124434	0.00870089	1.38123e-11
38	124434	0.00919799	1.38048e-11
39	124434	0.0097089	1.37958e-11
40	124434	0.0102336	1.37924e-11
41	124434	0.0107722	1.37893e-11
42	124434	0.0113245	1.37816e-11
43	124434	0.0118906	1.37766e-11
44	124434	0.0124706	1.37693e-11
45	124434	0.0130644	1.37604e-11
46	124434	0.013672	1.37526e-11
47	124434	0.0142933	1.37445e-11
48	124434	0.0149285	1.37362e-11
49	124434	0.0155776	1.37252e-11
50	124434	0.0162404	1.37189e-11
51	124434	0.016917	1.37072e-11
52	124434	0.0176074	1.36978e-11
53	124434	0.0183117	1.36867e-11
54	124434	0.0190298	1.36723e-11
55	124434	0.0197616	1.36644e-11
56	124434	0.0205073	1.36523e-11
57	124434	0.0212668	1.36425e-11
58	124434	0.0220401	1.36271e-11
59	124434	0.0228272	1.3615e-11
60	124434	0.0236282	1.36015e-11
61	124434	0.0244429	1.35878e-11
62	124434	0.0252714	1.35764e-11
63	124434	0.0261138	1.35585e-11
64	124434	0.02697	1.35427e-11
65	124434	0.02784	1.35284e-11
66	124434	0.0287238	1.35107e-11
67	124434	0.0296214	1.34912e-11
68	124434	0.0305328	1.34764e-11
69	124434	0.031458	1.34539e-11
70	124434	0.0323971	1.3435e-11

71	124434	0.0333499	1.34169e-11
72	124434	0.0343166	1.33968e-11
73	124434	0.0352971	1.33769e-11
74	124434	0.0362914	1.33586e-11
75	124434	0.0372995	1.33359e-11
76	124434	0.0383214	1.33135e-11
77	124434	0.0393572	1.32904e-11
78	124434	0.0404067	1.32619e-11
79	124434	0.0414701	1.32445e-11
80	124434	0.0425473	1.3215e-11
81	124434	0.0436383	1.3192e-11
82	124434	0.0447431	1.31657e-11
83	124434	0.0458617	1.31392e-11
84	124434	0.0469941	1.31119e-11
85	124434	0.0481404	1.30853e-11
86	124434	0.0493005	1.30566e-11
87	124434	0.0504743	1.30247e-11
88	124434	0.051662	1.29952e-11
89	124434	0.0528636	1.29659e-11
90	124434	0.0540789	1.29356e-11
91	124434	0.055308	1.28983e-11
92	124434	0.056551	1.28713e-11
93	124434	0.0578078	1.28325e-11
94	124434	0.0590784	1.28031e-11
95	124434	0.0603628	1.27631e-11
96	124434	0.061661	1.27286e-11
97	124434	0.062973	1.26924e-11
98	124434	0.0642989	1.26567e-11
99	124434	0.0656386	1.26157e-11
100	124434	0.0669921	1.2576e-11

Elapsed cpu time for main computation:
27.253 seconds.

Test case 2, thread number=2:

Elapsed cpu time for main computation:
15.7965 seconds.

Test case 3, thread number=4:

Elapsed cpu time for main computation:
16.4121 seconds.

For jacobi.txt:

Test case 1, thread number=1:

JACOBI
  C++ version
  Number of threads = 1
  Problem size N = 500
        1   187276
        2   93285.9
        3   46462.4
        4   23141.2
        5   11525.7
        6   5740.53
        7   2859.14
        8   1424.03
        9   709.254
       10   353.253
       11   175.942
       12   87.6299
       13   43.6451
       14   21.738
       15   10.8269
       16   5.39244
       17   2.68577
       18   1.33768
       19   0.666248
       20   0.331833
       21   0.165273
       22   0.0823163
       23   0.0409986
       24   0.0204199
       25   0.0101704
       26   0.00506547
       27   0.00252292
       28   0.00125657
       29   0.000625849
       30   0.000311712
       31   0.000155252
       32   7.7325e-05
       33   3.85126e-05
       34   1.91817e-05
       35   9.55366e-06
       36   4.75831e-06
       37   2.36993e-06
       38   1.18037e-06
       39   5.879e-07

DIFF   = 5.879e-07
DIFF_TOL = 9.38289e-07
Time   = 0.10055

  First 10 elements of estimated solution:

0	1
1	2
2	3
3	4
4	5
5	6
6	7
7	8
8	9
9	10

Test case 2, thread number=2:

JACOBI
C++ version
Number of threads = 2
Problem size N = 500

1	187276
2	93285.9
3	23231
4	23141.2
5	5762.8
6	2870.23
7	1429.59
8	712.006
9	709.254
10	176.628
11	87.9698
12	87.6299
13	21.8228
14	21.738
15	5.41349
16	2.69625
17	1.34287
18	0.668848
19	0.33312
20	0.165914
21	0.0826357
22	0.0411577
23	0.0204995
24	0.0204199
25	0.0101704
26	0.00506547
27	0.00252292
28	0.000628277
29	0.000312921
30	0.000311712

```
   31    7.76249e-05
   32    3.86629e-05
   33    1.92565e-05
   34    1.91817e-05
   35    9.55366e-06
   36    4.75831e-06
   37    1.18495e-06
   38    5.90179e-07

DIFF     = 5.90179e-07
DIFF_TOL = 9.38289e-07
Time     = 0.0608654

  First 10 elements of estimated solution:

    0        1
    1        2
    2        3
    3        4
    4        5
    5        6
    6        7
    7        8
    8        9
    9       10

Test case 3, thread number=4:

JACOBI
  C++ version
  Number of threads = 4
  Problem size N = 500
    1      23384.6
    2      23333.3
    3      11616.4
    4      5785.86
    5      2881.13
    6      1435.02
    7       714.71
    8       355.97
    9      177.331
   10       88.3219
   11       43.9808
   12       21.9052
   13       10.9101
   14       5.43503
   15       2.70698
   16       1.34824
   17      0.671509
```

18	0.334453
19	0.166578
20	0.0829664
21	0.0413224
22	0.0205811
23	0.0102507
24	0.00510547
25	0.00254284
26	0.00126649
27	0.000630791
28	0.000314173
29	0.000156478
30	7.79356e-05
31	3.88161e-05
32	1.93292e-05
33	9.62715e-06
34	4.79581e-06
35	2.38817e-06
36	1.18945e-06
37	5.92424e-07

DIFF = 5.92424e-07
DIFF_TOL = 9.38289e-07
Time = 0.0779878

First 10 elements of estimated solution:

0	1
1	2
2	3
3	4
4	5
5	6
6	7
7	8
8	9
9	10