National Chiao Tung University
Department of Electrical Engineering
Computer Intelligence on Automation (C.I.A.) Lab

GEE9018 F16 Midterm
Nov 14, 2016
Prof. Hung-Pin (Charles) Wen

GEE9018 (5095) F'16 Midterm: Programming Examination

TIME: 2016/11/17 15:30 – 18:20

Problem01 (50%) Please use OpenMP to parallelize the following problems:

(a) (20%) Calculating pi(π) with Monte Carlo

The Monte-Carlo method is a random sampling technique and known to be applicable to solve the problem of calculating pi(π), in which a target circle ($x^2 + y^2 = 1$) with radius R = 1 is used. The Monte-Carlo method then randomly chooses a pair (x, y), both ranging from 0 to1, in the first quadrant as shown in Figure 1.

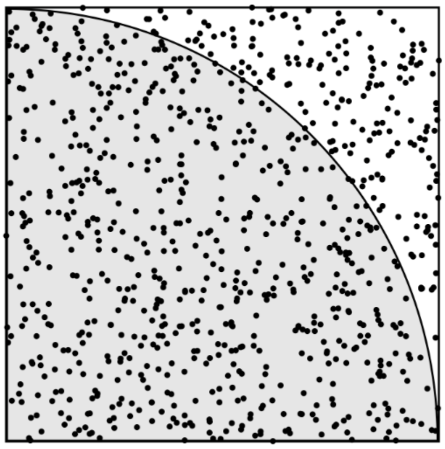

Fig 1. Circle area = π/4

As a result, the relationship between the area and random particles can be expressed as:

$$\frac{\pi}{4} = \frac{\#particles\ \ in\ \ gray\ \ area}{\#total\ \ particles}$$

Note: You must use 1 as the seed in your modified program once the library routine for randomization is called. Please complete/compile your program (with a designated thread count 8) and run as

```
> ./P1a_omp 1000000000
PI = 3.141520
time: 3.442488 s
```

(b) (30%) Finding the inverse matrix with Gauss-Jordan elimination

Given a sample matrix represented as follows,

$$\left[\begin{array}{ccc|ccc} a_{11} & a_{12} & a_{13} & 1 & 0 & 0 \\ a_{21} & a_{22} & a_{23} & 0 & 1 & 0 \\ a_{31} & a_{32} & a_{33} & 0 & 0 & 1 \end{array}\right] = \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & x_{11} & x_{21} & x_{31} \\ 0 & 1 & 0 & x_{12} & x_{22} & x_{32} \\ 0 & 0 & 1 & x_{13} & x_{23} & x_{33} \end{array}\right]$$

where A is the target matrix, and an identity matrix is augmented after A. Gauss-Jordan elimination is performed on this matrix iteratively until the left-hand side of the augmented matrix becomes an identity matrix. At the end, the matrix X is the result for the inverse of matrix A.

Note 1: Do not print the final result on screen (only use when debugging).

Note 2: Six test cases (from input1.txt to input6.txt) are provided.

Note 3: Use command “diff” to verify your result with the golden answer from the serial program.

Please complete/compile your program (with a designated thread count 8) and run as

```
(serial program)
> ./P1b input5.txt output5_golden.txt
Memory usage: 8672 k-bytes
time: 3.709641 s

(parallel program)
> ./P1b_omp input5.txt output5.txt
Memory usage: 8324 k-bytes
time: 1.070326 s

(verify the results)
> diff output5.txt output5_golden.txt
```

Problem02 (50%) Please use the POSIX pthreads to parallelize the following problem: Brownian motion is the random motion of particles suspended in a fluid. The motion of pollens will be observed to prove the random motion of particles indirectly by simulation, in which the movement of a piece of pollen (a.k.a. a "big particle") is targeted.

"p2_pth.c" and "B_Motion.h" are the two files used for this problem. Please type "make" on screen to compile your code and output your executable program as "p2_pth".

Please note that "process()" is the recommended (but not limited) function suitable for parallelization and mainly consists of three steps per timestep:

1. move(): move the particles first.
2. get_distance(): calculate the distances between every two particles.
3. collision(): evaluate if collision occurs between two particle and then update speeds and directions of particles if collision happens.

Be aware of "pos" in each particle when updating the position, "distance_list" in each particle when updating the distances, and "velocity" in each particle when updating its speed. You may refer to "B_Motion.h" for more details if necessary.

The basic information and the movement of the pollen will be printed on screen:

```
> ./p2_pth 1 3

=======   Brownian Motion Simulation   =======
Thread number is 1.
Temperature is 70 degree.
Space is 3.000 (in 0.0001 * micrometer^3)
Number of time step is 10 ( per 1e-11 sec )

Environment Status:
            Number of gas particles: 2744
            Kinetic energy of each gas particle: 7.100100e-21 (J)
            Velocity of each gas particle: 17.472959 (m/s)
            Mass of a gas particle: 4.651163e-23 (kg)
            Mass of the pollen: 4.651163e-21 (kg)

Step: 0     x: 3.347165e-08     y: 3.586248e-08     z: 3.347165e-08
```

```

Step: 1     x: 3.360165e-08     y: 3.595248e-08     z: 3.336165e-08

Step: 2     x: 3.348165e-08     y: 3.595248e-08     z: 3.342165e-08

Step: 3     x: 3.353165e-08     y: 3.597248e-08     z: 3.343165e-08

Step: 4     x: 3.350165e-08     y: 3.598248e-08     z: 3.342165e-08

Step: 5     x: 3.346165e-08     y: 3.595248e-08     z: 3.342165e-08

Step: 6     x: 3.346165e-08     y: 3.595248e-08     z: 3.340165e-08

Step: 7     x: 3.346165e-08     y: 3.596248e-08     z: 3.340165e-08

Step: 8     x: 3.346165e-08     y: 3.596248e-08     z: 3.340165e-08

Step: 9     x: 3.346165e-08     y: 3.596248e-08     z: 3.340165e-08
Memory usage: 59860 bytes
Time: 2.620244 sec
```

The movement of the pollen will be printed and plotted in file “answer.txt”.

```
> cat answer.txt
Step: 0     x: 3.347165e-08     y: 3.586248e-08     z: 3.347165e-08
Step: 1     x: 3.360165e-08     y: 3.595248e-08     z: 3.336165e-08
Step: 2     x: 3.348165e-08     y: 3.595248e-08     z: 3.342165e-08
Step: 3     x: 3.353165e-08     y: 3.597248e-08     z: 3.343165e-08
Step: 4     x: 3.350165e-08     y: 3.598248e-08     z: 3.342165e-08
Step: 5     x: 3.346165e-08     y: 3.595248e-08     z: 3.342165e-08
Step: 6     x: 3.346165e-08     y: 3.595248e-08     z: 3.340165e-08
Step: 7     x: 3.346165e-08     y: 3.596248e-08     z: 3.340165e-08
Step: 8     x: 3.346165e-08     y: 3.596248e-08     z: 3.340165e-08
Step: 9     x: 3.346165e-08     y: 3.596248e-08     z: 3.340165e-08
===================== Plot ====================
                       O
                  O    O
                         O

                        O


                       O








====================================================
Memory usage: 59860 bytes
Time: 2.620244 sec
```

Problem03 (20%) Please use the task structure in OpenMP for parallelization to modify the source program (P3_omp.c) that solves a recursive function for binomial coefficients, expressed as

$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k} \quad \text{for all integers } n,k : 1 \leq k \leq n-1,$$

with initial/boundary values

$$\binom{n}{0} = \binom{n}{n} = 1 \quad \text{for all integers } n \geq 0$$

```
>./P3_omp 5 2
Please input number n and k:5 2
C(5,2)=10
>
```