并行计算 HW4

PB19071501 李平治

15.1

1.

该代码的功能是双进程并行计算快速傅立叶变换

2.

会使得该函数一直阻塞不返回，从而导致程序无法中止

15.3

```
float data[1024], buff[10];
for (int i=0; i<10; i++) buff[i] = data[32*i];
MPI_Send(buff, 10, MPI_FLOAT, dest, tag, MPI_COMM_WORLD);
```

15.13.(1)

C++代码:

```
#include <iostream>
#include <cmath>
#include <ctime>
#define N 1000000
using namespace std;

static double random_double_in_range(double max) {
    return static_cast <double> (rand()) / (static_cast <double> (RAND_MAX/max));
}

```

```cpp
static double d2r(double d) {
  return d / 180.0 * ((double) M_PI);
}

bool random_toss_neddle(double l, double a, double b){
    double x = random_double_in_range(10000);
    double y = random_double_in_range(10000);
    double angle = random_double_in_range(360.0);
    double x_ = x + l * cos(d2r(angle));
    double y_ = y + l * sin(d2r(angle));
    return ( (int)(x/a) != (int)(x_/a) ) or ( (int)(y/b) != (int)(y_/b) );
}

int main(){
    double l = 1, a = 1, b = 1;
    double pi;
    int log_step = 10000;
    int total_count, edge_count = 0;
    auto start_time = clock();
    for(int i=0; i<N; i++){
        total_count += 1;
        if (random_toss_neddle(l, a, b)) {
            edge_count += 1;
        }
        pi = 3.0 * (double)total_count / (double)edge_count;
        if (i % log_step == 0) {
            cout << "Step " << i << ", estimated Pi: " << pi << endl;
        }
    }
    auto end_time = clock();
    cout << "runtime: " << (double)(end_time - start_time) / CLOCKS_PER_SEC <<
endl;
    return 0;
}
```

运行结果：

```
Step 100000, estimated Pi: 3.14515
Step 200000, estimated Pi: 3.14505
Step 300000, estimated Pi: 3.14285
Step 400000, estimated Pi: 3.14272
Step 500000, estimated Pi: 3.14246
Step 600000, estimated Pi: 3.14249
Step 700000, estimated Pi: 3.14294
Step 800000, estimated Pi: 3.14299
Step 900000, estimated Pi: 3.14235
runtime: 0.093622
```

运行模拟时间为$0.093622s$，π 精确位数为3位