高性能并行计算第 2 次作业

姓名: 张子栋 学号: 2020317210101

代码地址:/home/2020317210101/work3

https://github.com/Bluuur/MarkdownNotes/tree/main/高性能并行计算/Code3

题目 1

使用 MPI 广播改写 Pi 计算程序

```
1 //
 2 // Created by ZidongZh on 2022/10/6.
   //
 4
 5
    #include<stdio.h>
 6
    #include<mpi.h>
 8
    int main(int argc, char *argv[]) {
 9
        double x;
        double sum;
10
        double h;
11
        double pi;
12
13
       double tick;
14
        double tack;
15
        int rank;
16
        int size;
17
18
        long long n;
        MPI_Init(&argc, &argv);
19
20
        MPI_Comm_rank(MPI_COMM_WORLD, &rank);
21
        MPI_Comm_size(MPI_COMM_WORLD, &size);
22
23
        n = 0;
24
        if (0 == rank) {
            printf("Please enter n:");
25
26
            scanf("%d", &n);
            tick = MPI_Wtime();
27
28
        }
29
30
        MPI_Bcast(&n, 1, MPI_INT, 0, MPI_COMM_WORLD);
31
```

```
h = 1.0 / (double) n;
32
        sum = 0.0;
33
34
        int i;
35
        for (i = rank + 1; i \le n; i += size) {
36
            x = h * ((double) i - 0.5);
37
            sum += (4 / (1 + x * x));
38
39
        }
        sum *= h;
40
41
        MPI_Reduce(&sum, &pi, 1, MPI_DOUBLE, MPI_SUM, 0,
42
    MPI_COMM_WORLD);
43
        if (rank == 0) {
44
45
            tack = MPI_Wtime();
            printf("Pi: %0.20f\ntime cost:%fs\n", pi, tack -
46
    tick);
47
        }
        return 0;
48
49
   }
```

运行结果

Pi: 3.14159265359042638721 time cost:1.674995s

题目 2

reduce

```
1 //
 2 // Created by ZidongZh on 2022/10/6.
 3
   //
 4
   #include<stdio.h>
   #include "mpi.h"
6
 7
   int main(int argc, char *argv[]) {
8
9
        static int NUM_STEPS = 1000000000;
10
       int rank;
11
       int size;
12
13
14
       double pi;
       double step;
15
```

```
double x;
16
17
        double sum;
        double start;
18
        double finish;
19
20
        MPI_Init(&argc, &argv);
21
        MPI_Comm_rank(MPI_COMM_WORLD, &rank);
22
23
        MPI_Comm_size(MPI_COMM_WORLD, &size);
        MPI_Bcast(&NUM_STEPS, 1, MPI_INT, 0, MPI_COMM_WORLD);
24
25
        step = 1.0 / (double) NUM_STEPS;
26
27
        sum = 0.0;
28
29
        if (rank == 0) {
30
            start = MPI_Wtime();
        }
31
32
33
        int i;
        for (i = rank; i < NUM_STEPS; i += size) {</pre>
34
            x = step * ((double) i - 0.5);
35
            sum += 4.0 / (1.0 + x * x);
36
37
        }
38
39
        sum = step * sum;
40
        MPI_Reduce(&sum, &pi, 1, MPI_DOUBLE, MPI_SUM, 0,
41
    MPI_COMM_WORLD);
42
43
        if (rank == 0) {
44
            finish = MPI_Wtime();
            printf("Pi: %.20f\nTime cost: %fs\n", pi, finish -
45
    start);
        }
46
47
        MPI_Finalize();
48
49
       return 0;
50
   }
51
```

运行结果

Pi: 3.14159265558997091716 Time cost: 16.747865s