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/work4`
`https://github.com/Bluuur/MarkdownNotes/tree/main/高性能并行计算/Code4`

题目: 列表线程数 2, 进程数 2, 4, 6, 8 的计算时间

代码:

```
#include<stdio.h>
#include"omp.h"
#include"mpi.h"
#define OMP_NUM_THREADS 2

int main(int argc, char *argv[]) {
    int myRank;
    int mySize;
    long numSteps = 1000000000;
    int low;
    int up;
    double x;
    double sum = 0.0;
    double pi = 0.0;
    double step = 1.0 / (double) numSteps;
    double tick;
    double tack;

    MPI_Status status;
    MPI_Init(&argc, &argv);
    MPI_Comm_rank(MPI_COMM_WORLD, &myRank);
    MPI_Comm_size(MPI_COMM_WORLD, &mySize);

    tick = MPI_Wtime();
    low = myRank * (numSteps / mySize);
    up = low + numSteps / mySize - 1;
    omp_set_num_threads(OMP_NUM_THREADS);
#pragma omp parallel for reduction(+:sum) private(x)
    for (int i = low; i < up + 1; i++) {
        x = (i - 0.5) * step;
        sum += 4.0 / (1.0 + x * x);
    }
```

```
33
34      MPI_Reduce(&sum, &pi, 1, MPI_DOUBLE, MPI_SUM, 0,
    MPI_COMM_WORLD);
35
36      if (myRank == 0) {
37          printf("Pi:%.12f\n", pi * step);
38      }
39      tack = MPI_Wtime();
40
41      if (myRank == 0) {
42          printf("time cost:%fs\n", tack - tick);
43      }
44      MPI_Finalize();
45
46      return 0;
47  }
48
```

进程数	Time/s	加速比	并行效率
1	16.752725	-	-
2	8.380567	1.998997	99.95%
4	4.213796	3.975685	99.39%
6	2.853538	5.870861	97.85%
8	2.150142	7.791450	97.39%