**并行计算 作业**

|  |  |  |  |
| --- | --- | --- | --- |
| 姓名 | 王少博 | 学号 | 181110315 |
| 总分 | 10 | 实际得分 |  |
| 作业内容（问题，思路，程序，结果，过程中遇到问题的解决方法） | | | |
| **问题：编写一个MPI程序，使用蒙特卡洛法估计的值**   1. 思路   通过进程0读入总的投掷数，然后使用MPI\_Send发送到其他进程，其他进程使用MPI\_Recv接收并计算pi值，再将pi值通过MPI\_Send发送回来。进程0使用MPI\_Recv接收。进程0读入总数，并且广播到其他进程。使用MPI\_Reduce获得局部变量number\_in\_circle的总和，由进程0打印结果。   1. 程序实现   #include <stdio.h>  #include <stdlib.h>  #include <string.h>  #include <mpi.h>  #include <time.h>  typedef long long ll; int main(void) {   srand((unsigned)time(NULL)); //随机数种子   long long int number\_of\_tosses = 1000000;//投掷次数   double pi\_estimate = 0;   ll int number\_in\_circle = 0;//在线程中计算的在圆内的点数   ll int total\_num\_in\_circle = 0;//在圆内总共的点数   int comm\_sz;   int my\_rank;   MPI\_Comm comm;   comm = MPI\_COMM\_WORLD;   //初始化   MPI\_Init(NULL, NULL);   MPI\_Comm\_size(MPI\_COMM\_WORLD, &comm\_sz);   MPI\_Comm\_rank(MPI\_COMM\_WORLD, &my\_rank);   printf("comm\_sz=%d\n",comm\_sz);   printf("my\_rank=%d\n", my\_rank);   //进程0   if (my\_rank == 0) {   printf("process 0\n");     for (int dest = 1; dest < comm\_sz; dest++) {   MPI\_Send(&number\_of\_tosses, 1, MPI\_LONG\_LONG\_INT, dest, 0, MPI\_COMM\_WORLD);//发送投掷次数   }     for (int src = 1; src < comm\_sz; src++) {   MPI\_Recv(&pi\_estimate, 1, MPI\_DOUBLE, src, 0, MPI\_COMM\_W ORLD, MPI\_STATUS\_IGNORE);   pi\_estimate += pi\_estimate;   }   printf("pi\_estimate=%lf\n", pi\_estimate/comm\_sz);   }   **//进程1**   else {   printf("process 1\n");   MPI\_Recv(&number\_of\_tosses, 1, MPI\_LONG\_LONG\_INT, 0, 0, MPI\_COMM\_WORLD, MPI\_STATUS\_IGNORE);   for (long long int toss = 0; toss < number\_of\_tosses; toss++) {   double x = rand() % 2 - 1;   double y = rand() % 2 - 1;   double distance\_squared = x \* x + y \* y;   if (distance\_squared <= 1)   number\_in\_circle++;   }   pi\_estimate = 4 \* number\_in\_circle / ((double)number\_of\_tosses);   MPI\_Send(&pi\_estimate, 1, MPI\_DOUBLE, 0, 0, MPI\_COMM\_WORLD);   }   MPI\_Finalize();   return 0;  }   1. 结果   投掷的次数比较小的时候，计算的值误差比较大，但是当我们投掷的次数增加之后，大概在1e8左右的时候，计算出的pi值就有点接近3.14了。因为越大的时候越容易收敛，bias越小。     1. 问题的解决方法   一开始使用ubuntu进行运行，总是报错，后来换成了windows之后就成功了，可能与线程库有关。 | | | |
| 教师评价 | | | |
|  | | | |