**高性能并行计算第1次作业**

姓名：代宏刚 学号：2020317110061

**代码地址：**

1. #include <stdio.h>
2. #include <time.h>
3. **long** num\_steps;
4. **double** step;
5. **int** main(){
6. printf("输入num\_steps:");
7. scanf("%d",&num\_steps);
8. **clock\_t** start,end;
9. **int** i;
10. **double** x,pi,sum = 0.0;
11. start = clock();
12. step = 1.0/(**double**)num\_steps;
13. **for**(i=1;i<=num\_steps;i++){
14. x = (i-0.5)\*step;
15. sum += 4.0/(1.0+x\*x);
16. }
17. pi = step \* sum;
18. end = clock();
19. printf("pi=%f\tclock=%.3lf\n",pi,(**double**)(end-start)/CLOCKS\_PER\_SEC);
20. **return** 0;
21. }

**实验结果：**

（实验结果一般使用表格展示，并对结果进行分析）

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| num\_steps | 100000 | 500000 | 1000000 | 10000000 | 100000000 | 500000000 | 1000000000 |
| time(s) | 0.004 | 0.011 | 0.019 | 0.169 | 1.678 | 8.386 | 16.745 |
| clock(s) | 0.000 | 0.000 | 0.010 | 0.160 | 1.670 | 8.380 | 16.74 |

分析：

表格第一行表示num\_steps的不同取值，第二行为time命令计算的程序执行时间，其值为user与sys时间之和，第三行为C的库函数clock计算的程序执行时间。从结果可发现，当num\_steps 倍数增长时，程序执行时间大致以相同的倍数增长。clock函数由于精度问题，在num\_steps比较小时，无法测出程序执行时间。