

小作业零 实验报告

吴泽文 2021013394

代码报告

part 1: openmp_pow.cpp 源代码

```
1 void pow_a(int *a, int *b, int n, int m) {
2     #pragma omp parallel for
3     for (int i = 0; i < n; i++) {
4         int x = 1;
5         for (int j = 0; j < m; j++)
6             x *= a[i];
7         b[i] = x;
8     }
9 }
```

part 2: mpi_pow.cpp 源代码

```
1 void pow_a(int *a, int *b, int n, int m, int comm_sz /* 总进程数 */) {
2     for (int i = 0; i < n / comm_sz; i++) {
3         int x = 1;
4         for (int j = 0; j < m; j++)
5             x *= a[i];
6         b[i] = x;
7     }
8 }
```

性能分析

见表格

prag (mpicxx mpi_pow.cpp -O3 -std=c++11 -o mpi_pow)	用时	加速比
openmp_pow: n = 112000, m = 100000, thread_count = 1	14003987 us	1
openmp_pow: n = 112000, m = 100000, thread_count = 7	2019510 us	6.93
openmp_pow: n = 112000, m = 100000, thread_count = 14	1009926 us	13.87
openmp_pow: n = 112000, m = 100000, thread_count = 28	510216 us	27.44
mpi_pow: n = 112000, m = 100000, process_count = 1	14014445 us	1
mpi_pow: n = 112000, m = 100000, process_count = 7	2020781 us	6.94
mpi_pow: n = 112000, m = 100000, process_count = 14	1018858 us	13.76
mpi_pow: n = 112000, m = 100000, process_count = 28	502176 us	27.91
mpi_pow: n = 112000, m = 100000, process_count = 56	316507 us	44.28

