

小作业零：pow_a

1. `openmp_pow.cpp` 和 `mpi_pow.cpp` 中修改后函数 `pow_a` 的源代码。

`openmp_pow.cpp`:

```
1 void pow_a(int *a, int *b, int n, int m) {
2     // TODO: 使用 omp parallel for 并行这个循环
3     #pragma omp parallel for
4     for (int i = 0; i < n; i++) {
5         int x = 1;
6         for (int j = 0; j < m; j++)
7             x *= a[i];
8         b[i] = x;
9     }
10 }
```

`mpi_pow.cpp`:

```
1 void pow_a(int *a, int *b, int n, int m, int comm_sz /* 总进程数 */) {
2     // TODO: 对这个进程拥有的数据计算 b[i] = a[i]^m
3     int local_n = n / comm_sz;
4     for (int i = 0; i < local_n; i++) {
5         int x = 1;
6         for (int j = 0; j < m; j++) {
7             x *= a[i];
8         }
9         b[i] = x;
10    }
11 }
```

2. 对 `openmp` 版本，报告使用 1, 7, 14, 28 线程在 $n = 112000$, $m = 100000$ 下的运行时间，及相对单线程的加速比。

线程数	运行时间 (us)	加速比
1	14005383	1.00
7	2011009	6.96
14	1021626	13.71
28	510356	27.44

3. 对 MPI 版本，报告 1×1 , 1×7 , 1×14 , 1×28 , 2×28 进程（ $N \times P$ 表示 N 台机器，每台机器 P 个进程）在 $n = 112000$, $m = 100000$ 下的运行时间，及相对单进程的加速比。

进程数	运行时间 (us)	加速比
1x1	14014714	1.00
1x7	2016049	6.95
1x14	1025108	13.66
1x28	504580	27.76
2x28	351082	39.89