

1. 源代码

```
1 // openmp_pow.cpp
2 void pow_a(int *a, int *b, int n, int m) {
3     // TODO: 使用 omp parallel for 并行这个循环
4     #pragma omp parallel for
5     for (int i = 0; i < n; i++) {
6         int x = 1;
7         for (int j = 0; j < m; j++)
8             x *= a[i];
9         b[i] = x;
10    }
11 }
```

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

2. OpenMP运行时间及加速比

线程数	运行时间(μs)	加速比
1	14025535	0.99
7	2024218	6.93
14	1011548	13.87
28	510098	27.50

- $S(p) \approx p$: 接近线性加速比

3. MPI版本运行时间及加速比

机器&进程	运行时间(μs)	加速比
1×1	14006121	1.00
1×7	2022122	6.93
1×14	1016888	13.77
1×28	529154	26.47
2×28	351049	39.90

- $S(p) \approx p$: 接近线性加速比
- 消息传递模型由于指令流之间基于通信同步, 所以加速效果低于OpenMP
 - 进程(线程)数相同的情况下, MPI的加速比一般低于OpenMP
 - 多机相对于单机只加速了1.51倍