实验零：`pow_a`

`openmp_pow.cpp` 函数 `pow_a`

```cpp
void pow_a(int *a, int *b, int n, int m) {
    // TODO: 使用 omp parallel for 并行这个循环
    #pragma omp parallel for
    for (int i = 0; i < n; i++) {
        int x = 1;
        for (int j = 0; j < m; j++)
            x *= a[i];
        b[i] = x;
    }
}
```

`mpi_pow.cpp` 函数 `pow_a`

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

`openmp` 版本

运行时间及相对单线程加速比如下表：

线程数	运行时间	加速比
1	$7757402\,\mu s$	1.00
7	$1308056\,\mu s$	5.93
14	$686286\,\mu s$	11.30
28	$339929\,\mu s$	22.82

`MPI` 版本

运行时间及相对单进程加速比如下表：

进程数	运行时间	加速比
1×1	$7671707\,\mu s$	1.00
1×7	$1350002\,\mu s$	5.68
1×14	$742070\,\mu s$	10.34
1×28	$387416\,\mu s$	19.80
2×28	$238340\,\mu s$	32.19
4×28	$138241\,\mu s$	55.50