

# 中山大学计算机学院本科生实验报告

课程名称：超级计算机原理与操作

任课教师：吴迪

年级	19	专业（方向）	计算机科学与技术
学号	19335174	姓名	施天予
开始日期	2021-4-10	完成日期	2021-4-10

## 一、实验题目

使用 openmp 实现并行版本的计数排序，参考代码中提供了串行版本和运行所需的主函数。

## 二、实验内容

排序的并行化书上讲了一种方法。一种是使用 `#pragma omp parallel for num_threads(thread_count)`，另一种是先用 `#pragma omp parallel num_threads(thread_count)`，再用 `#pragma omp for`，在这个实验中因为一共使用了一个并行循环，所以两者效果应该差不多，我选了第二种写法。

在并行的 `for` 循环中，`n`（规模）和 `a`（数组）还有 `temp`（临时数组）是每个线程共享的，所以设为共享变量；`i` 和 `j` 是循环变量，`count` 用于计数，是每个线程独立的，所以设为私有变量。

我发现串行计数排序中的 `memcpy(a, temp, n*sizeof(int))` 语句也可以并行化，用 `#pragma omp for` 写一个简单的赋值 `for` 循环即可。

## 三、实验结果

```
void Count_sort_parallel(int a[], int n, int thread_count) {
    int i, j, count;
    int* temp = malloc(n*sizeof(int));

    # pragma omp parallel num_threads(thread_count) \
        default(none) shared(n,a,temp) private(i,j,count)
    # pragma omp for
    for (i = 0; i < n; i++) {
        count = 0;
        for (j = 0; j < n; j++)
            if (a[j] < a[i])
                count++;
            else if (a[j] == a[i] && j < i)
                count++;
        temp[count] = a[i];
    }

    # pragma omp parallel for
    for (i = 0; i < n; ++i)
        a[i] = temp[i];
    free(temp);
}
```

编译运行（设置规模  $n = 10000$ ）

```
sty@ubuntu:~$ cd /home/sty/Parallel/OpenMP
sty@ubuntu:~/Parallel/OpenMP$ gcc -g -Wall -fopenmp -o hw4_openmp hw4_openmp.c
sty@ubuntu:~/Parallel/OpenMP$ ./hw4_openmp 4 10000
Serial run time: 4.468091e-01

Parallel run time: 1.468019e-01

qsort run time: 1.353979e-03
sty@ubuntu:~/Parallel/OpenMP$ ./hw4_openmp 8 10000
Serial run time: 4.511039e-01

Parallel run time: 1.007431e-01

qsort run time: 1.287937e-03

sty@ubuntu:~/Parallel/OpenMP$ ./hw4_openmp 10 10000
Serial run time: 4.471579e-01

Parallel run time: 8.541203e-02

qsort run time: 1.353979e-03
sty@ubuntu:~/Parallel/OpenMP$ ./hw4_openmp 100 10000
Serial run time: 4.489520e-01

Parallel run time: 9.998512e-02

qsort run time: 1.703978e-03
sty@ubuntu:~/Parallel/OpenMP$ ./hw4_openmp 1000 10000
Serial run time: 4.470930e-01

Parallel run time: 3.215752e-01

qsort run time: 1.481056e-03
```

运行程序发现，并行计数排序的速度要比串行计数排序快，但是比快速排序要慢。当线程个数为 4 时，并行版本可以比串行版本将近快 4 倍。但随着线程数的增加，速度与线程数不再呈正比，大约在线程数为 10 时达到最快。当线程数继续增加时，并行版本的速度反而会变慢，因为线程此时很可能被频繁切换，导致时间变慢。