

Задание 6

Отчёт

Параллельная сортировка слиянием с использованием OpenMP

Ши Хуэй shihuicollapsor@gmail.com

1. Постановка задачи

Целью данной задачи является реализация параллельной сортировки слиянием для массива целых чисел. Алгоритм включает три основных этапа:

1. Разбиение массива на чанки.
2. Сортировка каждого чанка параллельно с использованием любого алгоритма сортировки.
3. Параллельное слияние отсортированных чанков в единый упорядоченный массив.

2. Формат командной строки

```
gcc -fopenmp -std=c99 task6.c -o task6  
./task6
```

3. Спецификация системы

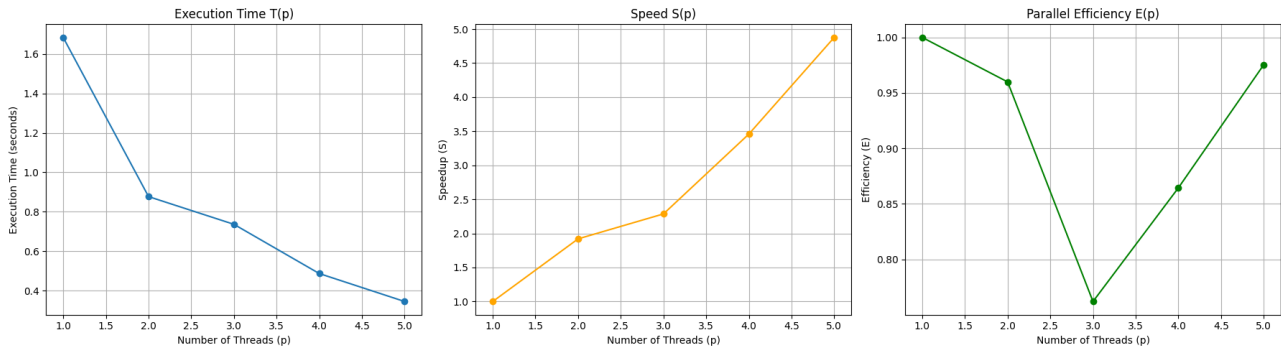
- Operating system : Linux 6.8.0-45-generic
- Vendor string and code : GenuineIntel (1, 0x1)
- Model string and code : Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz (165, 0xa5)

4. Записи экспериментов и результаты

4.2 Тестирование

Данный код реализует параллельный алгоритм сортировки слиянием для сортировки крупного массива. Вначале создается массив размером $N=100000000$, заполненный случайными числами, и его копия для последующей переинициализации. В главной функции массив сортируется с помощью функции `parallel_merge_sort`, которая рекурсивно делит массив на подмассивы и сортирует их параллельно. Если размер подмассива меньше заданного порога (`THRESHOLD=10000`), используется функция `qsort` для оптимизации.

В функции `parallel_merge_sort` массив делится на несколько частей, которые сортируются параллельно с использованием директивы OpenMP `#pragma omp task`, а затем объединяются с помощью функции `merge`. Код вычисляет время выполнения при разном количестве потоков ($P=1,2,4,8,16$), ускорение (S) и эффективность параллелизма (E), а также проверяет корректность сортировки через функцию `is_sorted`.



4.2 Результат вывода программы

Performance_Metrics_for_Different_Thread_Counts

P	Execution Time (T)	Speed (S)	Parallel Efficiency (E)
1	1.217016	1.000000	1.000000
2	0.876963	1.919847	0.959924
4	0.736485	2.286041	0.762014
8	0.486915	3.457762	0.864441
16	0.345224	4.876941	0.975388

```
collapsor@collapsor-G5-5500:~/Desktop/tspp_shihui/sem06$ ./task6
```

```
Now P = 1,Sorting correct!
```

```
Now P = 2,Sorting correct!
```

```
Now P = 4,Sorting correct!
```

```
Now P = 8,Sorting correct!
```

```
Now P = 16,Sorting correct!
```

```
Sum_List Execution time: 1.683635 0.876963 0.736485 0.486915 0.345224
```

```
Speed (S):1.000000 1.919847 2.286041 3.457762 4.876941
```

```
Parallel efficiency (E): 1.000000 0.959924 0.762014 0.864441 0.975388
```