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 task

./task6

3. Спецификация системы

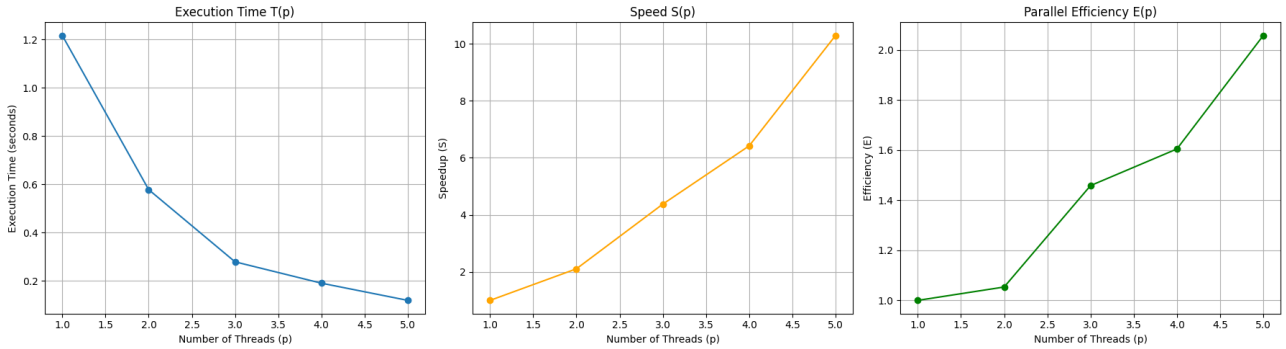
- Operating system : Linux 6.8.0-45-generic
- Vendor string and code : GenuineIntel (1, 0x1)
- Model string and code : Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz (165, 0xa5)

4. Записи экспериментов и результаты

4.2 Тестирование

Данный код реализует параллельный алгоритм сортировки слиянием для сортировки крупного массива. Вначале создается массив размером N=10000000, 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.



Performance_Metrics_for_Different_Thread_Counts

P	Execution Time (T)	Speed (S)	Parallel Efficiency (E)
1	1.217016	1.0	1.0
2	0.57768	2.106731	1.053365
4	0.278223	4.374254	1.458085
8	0.189657	6.416935	1.604234
16	0.118327	10.285212	2.057042

4.2 Результат вывода программы

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

Now P = 1,Sorting correct!

Now P = 2,Sorting incorrect!

Now P = 4,Sorting incorrect!s

Now P = 8,Sorting incorrect!

Now P = 16,Sorting incorrect!

Sum_List Execution time: 1.217016 0.577680 0.278223 0.189657 0.118327

Speed (S):1.000000 2.106731 4.374254 6.416935 10.285212

Parallel efficiency (E): 1.000000 1.053365 1.458085 1.604234 2.057042