



**Спецкурс: системы и средства параллельного
программирования.**

**Отчет по реализации параллельной сортировки
слиянием.**

Работу выполнил
Гайсин Р. Р.

Постановка задачи.

Задача: Реализовать параллельную сортировку слиянием с помощью OpenMP (либо с помощью программного интерфейса POSIX Threads). Программа должна принимать на вход целочисленные n , p , где p – число OpenMP потоков (либо число POSIX Threads, проводящих непосредственно сортировку чанков). Составить график зависимости $T(p)$, $S(p)$, $E(p)$ при фиксированном значении n . На графике $T(p)$ отобразить время работы qsort из `stdlib.h`.

Описание программы.

Описание основных функций: Программа принимает на вход два числа — размер массива и число нитей, на которых будет производиться сортировка. Затем генерируется целочисленный массив случайных значений и подается в функцию сортировки. В функции сортировки массив разбивается на части и каждая часть по отдельности сортируется отдельной нитью функцией `qsort`. Затем над отсортированными частями массива проводится процедура многопоточной сортировки слиянием.

Анализ времени выполнения: Для оценки времени выполнения программы использовалась функция `omp_get_wtime`. Для повышения надёжности экспериментов опыты проводились несколько раз - по десять итерации для каждого количества вызываемых для сортировки процессов, затем проводилось усреднение по времени

Результаты выполнения.

Результаты:

Проводилась сортировка массивов размером 10 000 000 элементов двумя способами — программой многопоточной сортировки и однопоточной программой, проводившей сортировку функцией `qsort`. Зависимость времени выполнения от числа нитей $T(p)$ представлена на графике (время в секундах), горизонтальная линия — среднее время эталонной однопоточной сортировки.

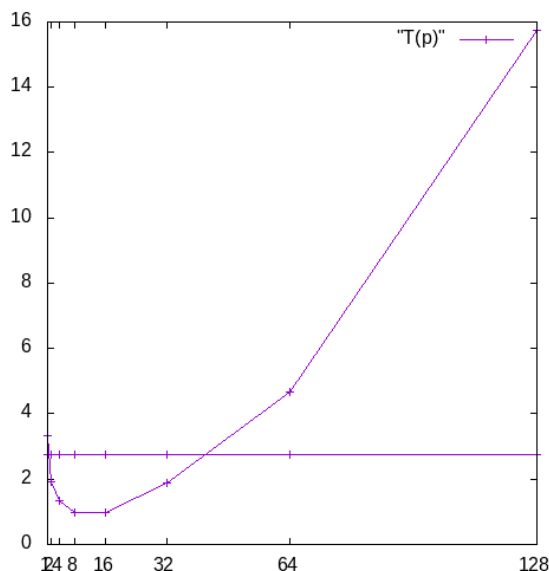


График зависимости времени выполнения от числа нитей

Ускорение $S(p)$ — отношение времени вычислений на одном потоке ко времени вычислений на p потоках представлено на следующем графике

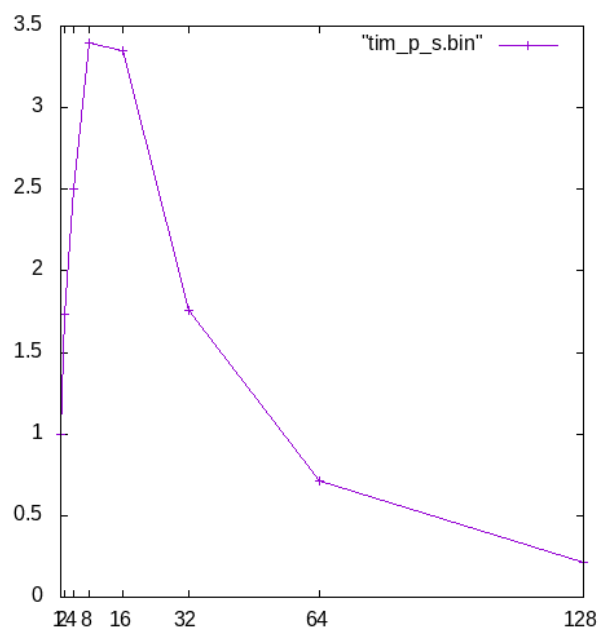


График ускорения $S(p)$ в зависимости от числа процессов

На следующем графике изображена зависимость эффективности параллельных вычислений E — это отношение полученного ускорения $S(p)$ к числу нитей p .

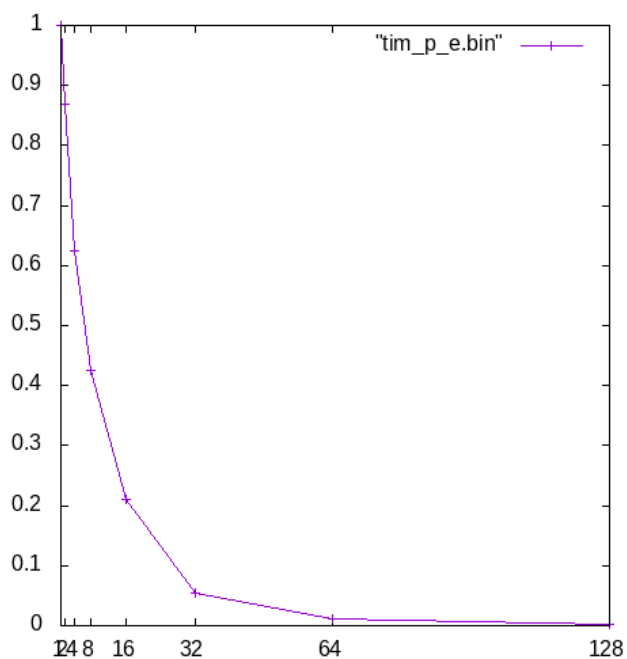


График зависимости эффективности параллельной программы от числа нитей

Основные выводы.

Исследования показывают, что при увеличении количества задействованных нитей время выполнения сортировки сперва снижается относительно времени сортировки однопоточной программы, затем начинает расти, так как увеличение числа потоков ведет к увеличению накладных расходов.