# Лабораторная работа 5. Параллелизм задач

**Задание**

На базе директив #pragma omp task реализовать многопоточный рекурсивный алгоритм быстрой сортировки (QuickSort). Опорным выбирать центральный элемент подмассива (функция partition, см. слайды к лекции). При достижении подмассивами размеров THREASHOLD = 1000 элементов переключаться на последовательную версию алгоритма.

Выполнить анализ масштабируемости алгоритма для различного числа сортируемых элементов и порогового значения THRESHOLD.

**Защита работы**

1.Продемонстрировать код программы и графики ускорения

2.Описать суть распараллеливания алгоритма

3.Охарактеризовать эффективность созданной параллельной программы

# Выполнение работы

Основная идея быстрой сортировки (quicksort)

Quicksort — это эффективный алгоритм сортировки, который работает по принципу "разделяй и властвуй". Алгоритм состоит из следующих шагов:

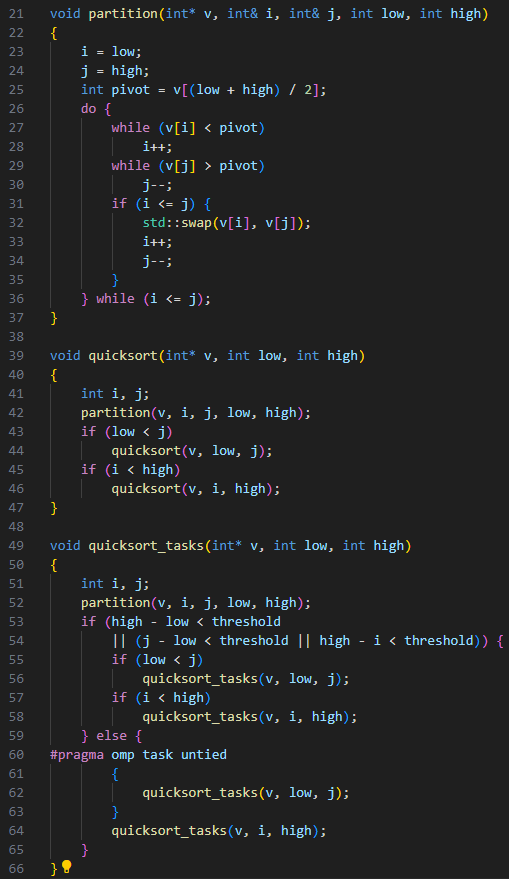
Выбор опорного элемента (pivot): Выбирается элемент из массива, называемый опорным.

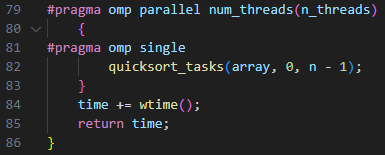
Разделение (partitioning): Массив разделяется на две части так, чтобы все элементы, меньшие опорного, находились слева, а все элементы, большие опорного, — справа.

Рекурсивная сортировка: Алгоритм рекурсивно применяется к двум подмассивам (левому и правому).

Параллельная версия алгоритма основывается на механизме создания и распределения задач Open MP. При этом существует порог (threshold) создания задачи, который назначается пользователем. Если размер подмассива больше заданного порога, то создаётся новая задача и помещается на один из потоков с дальнейшим распределениям по другим потокам из пула, а если же подмассив меньше порогового значения, то новая задача не создаётся и алгоритм продолжает своё выполнение внутри той задачи, в которой был рекурсивный вызов. Ключевое слово “untied” позволяет задаче быть выполненной любым потоком в пуле, что повышает гибкость выполнения.

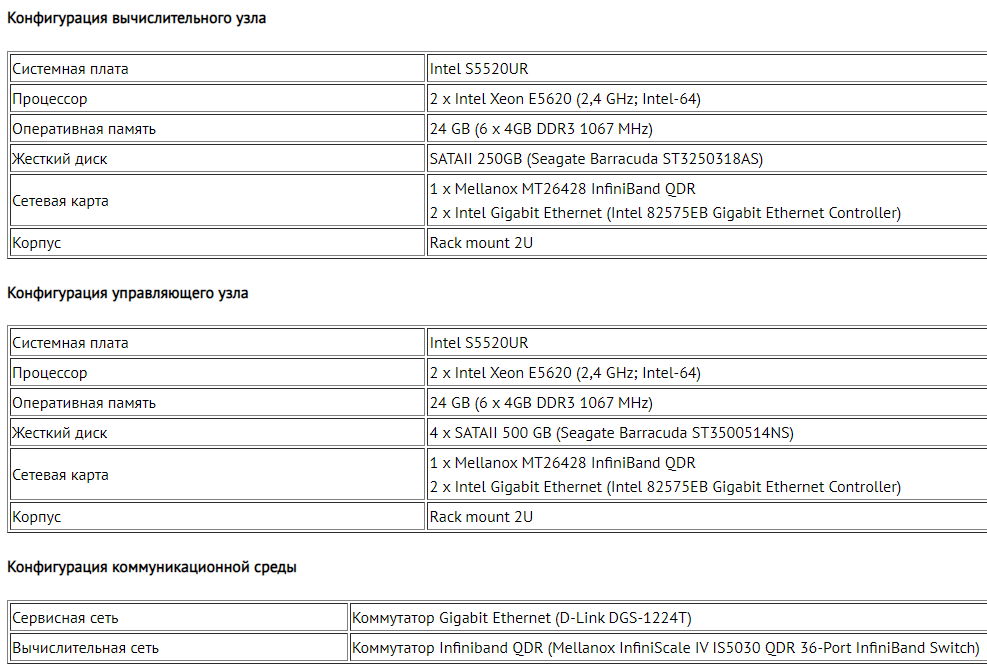
# Код алгоритма

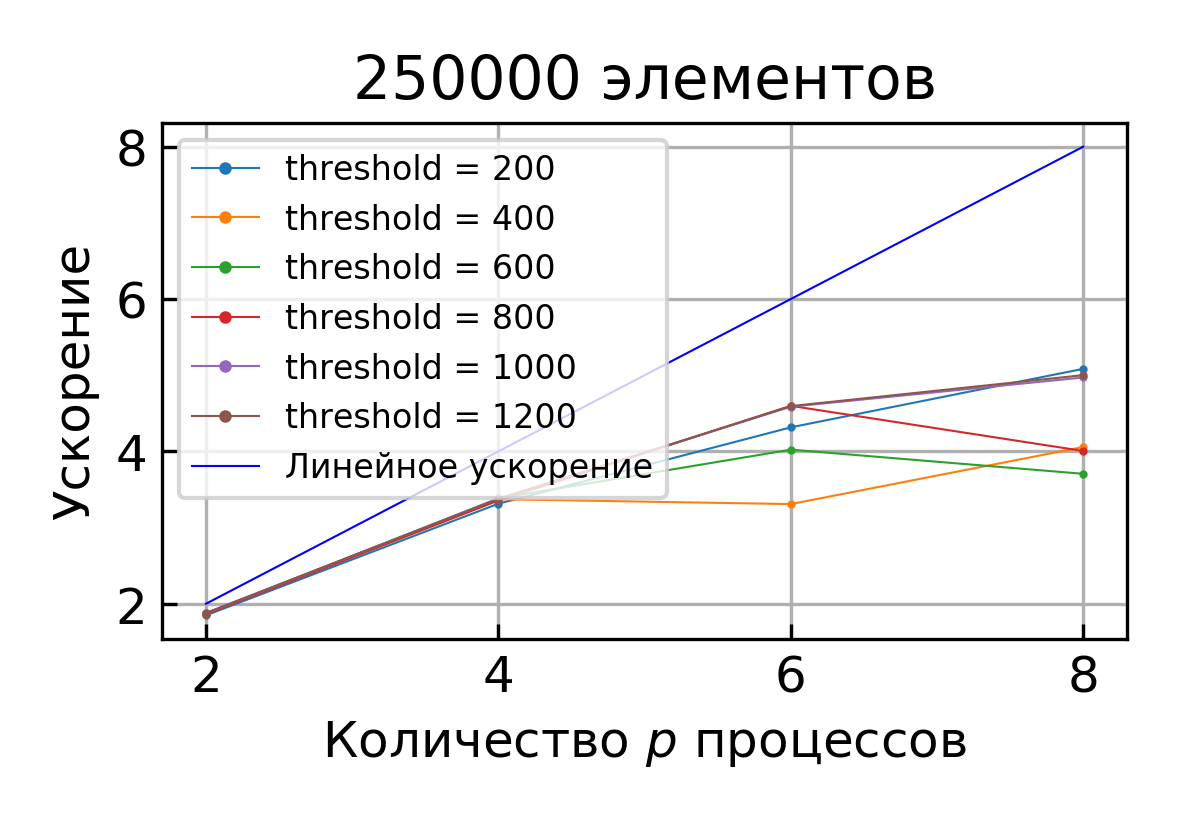


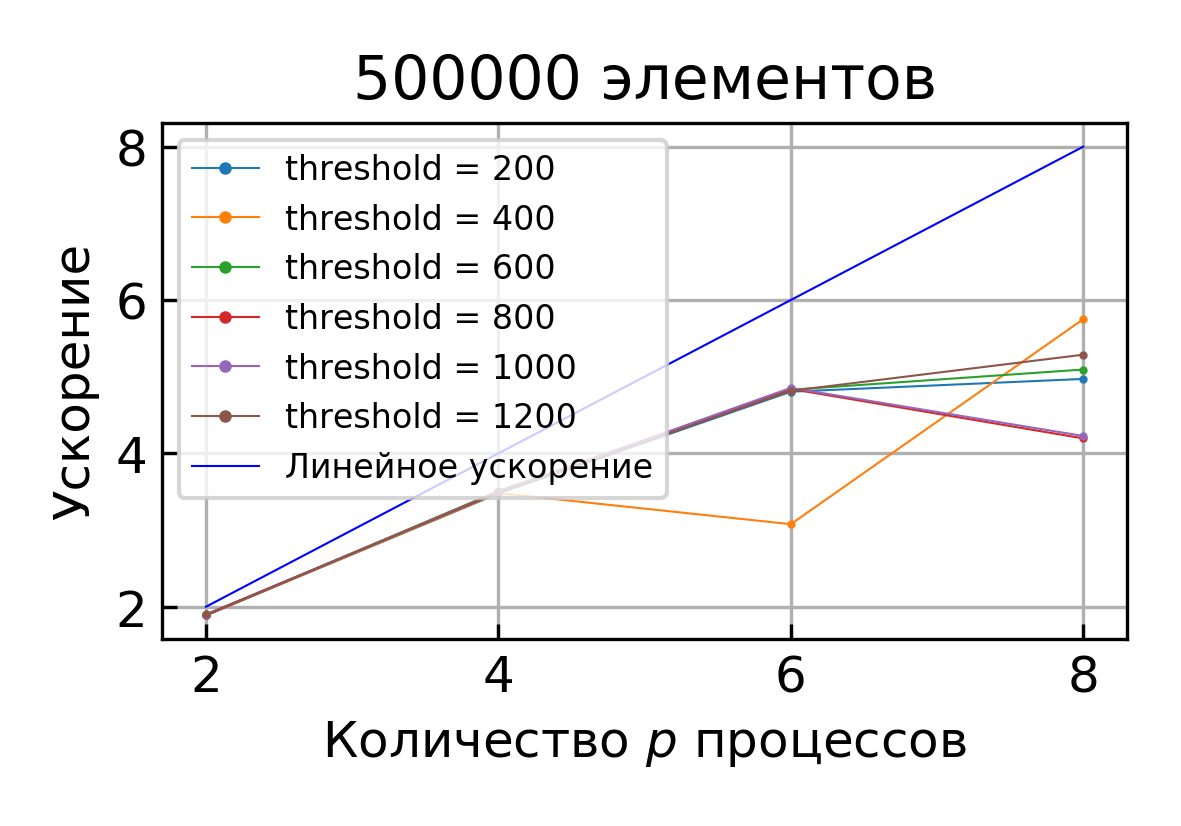


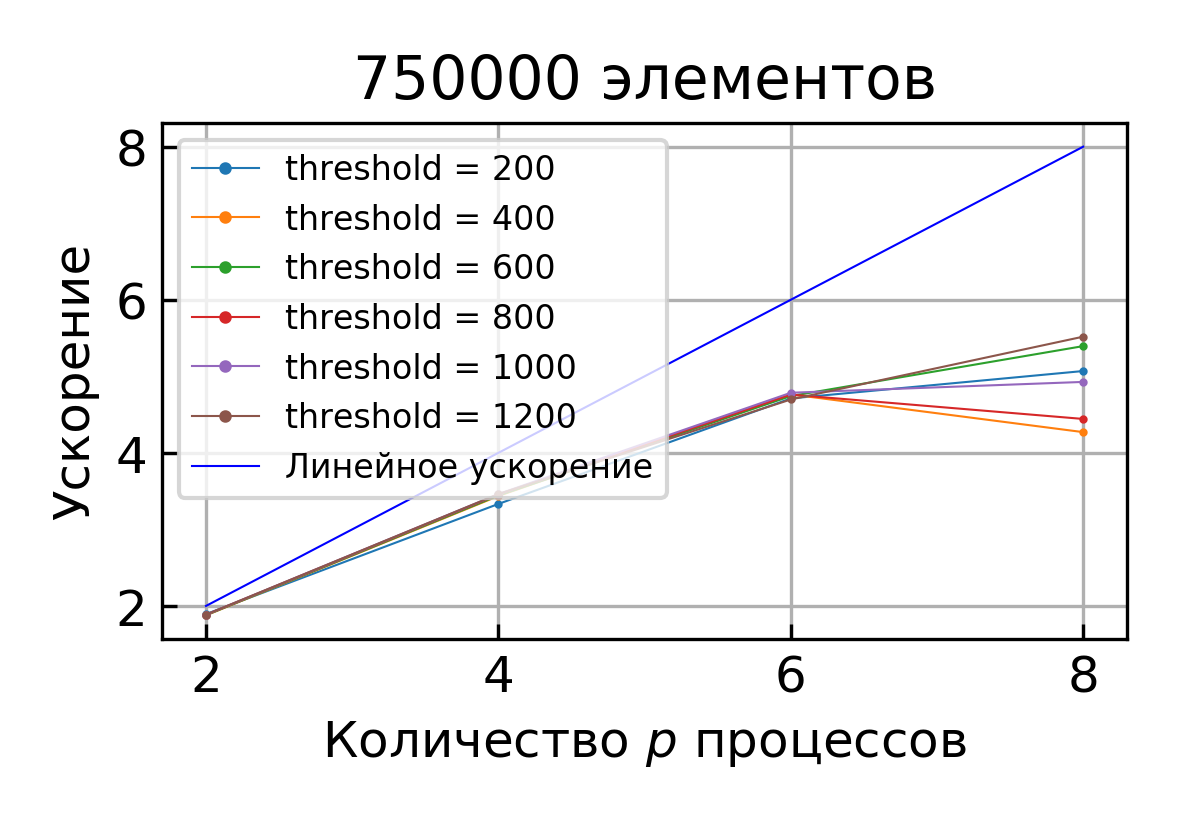
# Графики

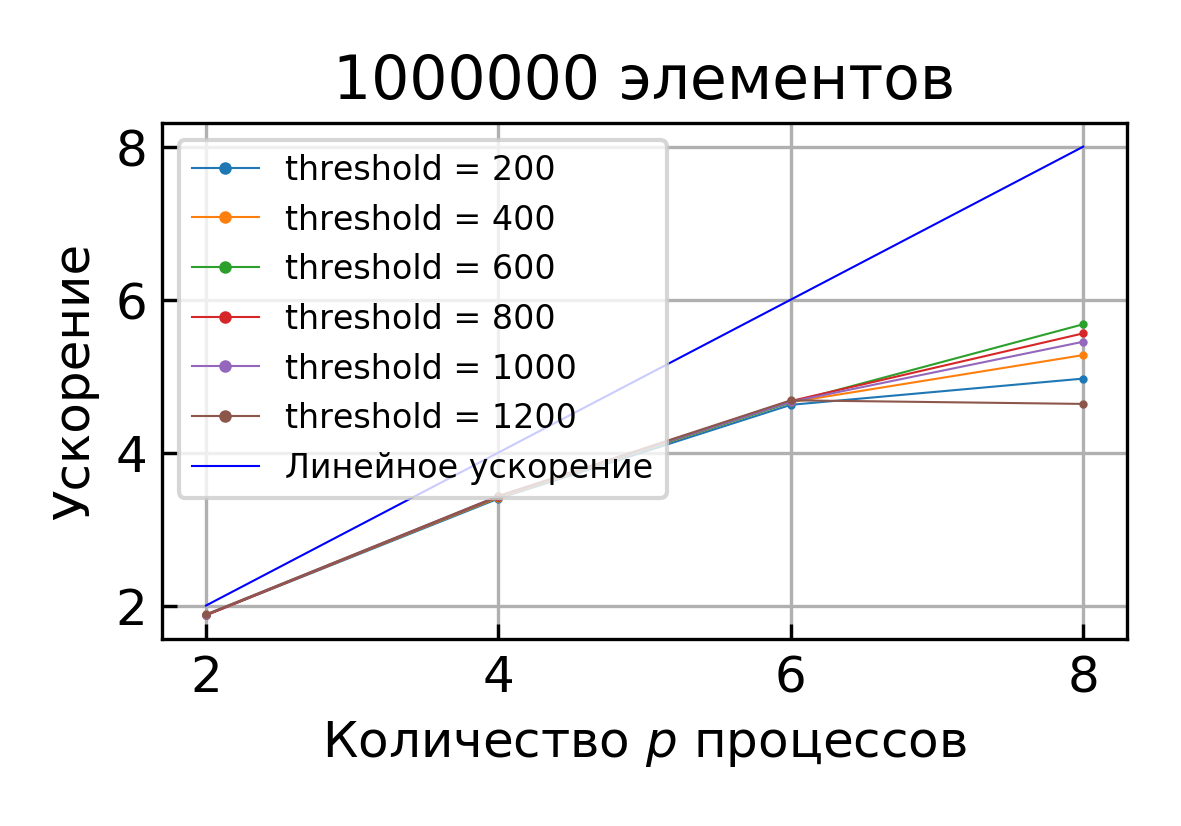
Все замеры производились на вычислительном кластере F (Oak) университета СибГУТИ.











# Заключение

Алгоритм показывает хорошее масштабирование на разных объёмах данный. Не трудно заметить, что масштабирование алгоритма отличается при разных пороговых значениях, поэтому имеет смысл подбирать оптимальный параметр для конкретной целевой системы.