

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 распределением 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

```
21 void partition(int* v, int& i, int& j, int low, int high)
22 {
23     i = low;
24     j = high;
25     int pivot = v[(low + high) / 2];
26     do {
27         while (v[i] < pivot)
28             i++;
29         while (v[j] > pivot)
30             j--;
31         if (i <= j) {
32             std::swap(v[i], v[j]);
33             i++;
34             j--;
35         }
36     } while (i <= j);
37 }
38
39 void quicksort(int* v, int low, int high)
40 {
41     int i, j;
42     partition(v, i, j, low, high);
43     if (low < j)
44         quicksort(v, low, j);
45     if (i < high)
46         quicksort(v, i, high);
47 }
48
49 void quicksort_tasks(int* v, int low, int high)
50 {
51     int i, j;
52     partition(v, i, j, low, high);
53     if (high - low < threshold
54         || (j - low < threshold || high - i < threshold)) {
55         if (low < j)
56             quicksort_tasks(v, low, j);
57         if (i < high)
58             quicksort_tasks(v, i, high);
59     } else {
60 #pragma omp task untied
61     {
62         quicksort_tasks(v, low, j);
63     }
64     quicksort_tasks(v, i, high);
65 }
66 }
```

```
79 #pragma omp parallel num_threads(n_threads)
80 {
81     #pragma omp single
82     quicksort_tasks(array, 0, n - 1);
83 }
84 time += wtime();
85 return time;
86 }
```

Графики

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

Конфигурация вычислительного узла

Системная плата	Intel S5520UR
Процессор	2 x Intel Xeon E5620 (2,4 GHz; Intel-64)
Оперативная память	24 GB (6 x 4GB DDR3 1067 MHz)
Жесткий диск	SATAII 250GB (Seagate Barracuda ST3250318AS)
Сетевая карта	1 x Mellanox MT26428 InfiniBand QDR 2 x Intel Gigabit Ethernet (Intel 82575EB Gigabit Ethernet Controller)
Корпус	Rack mount 2U

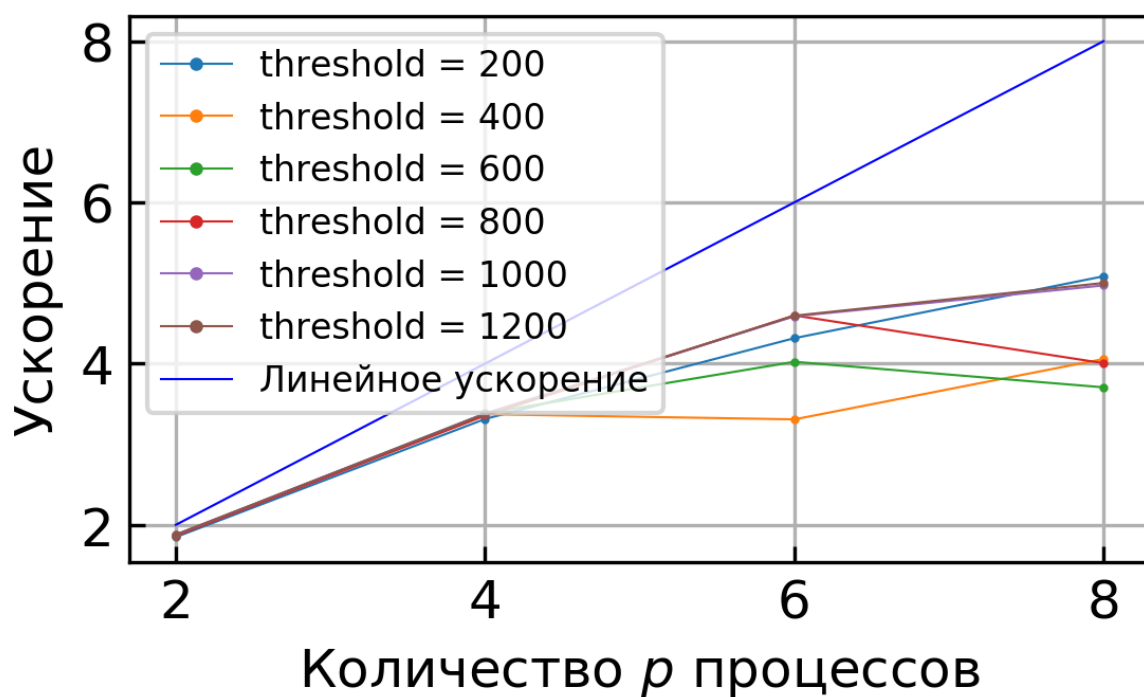
Конфигурация управляющего узла

Системная плата	Intel S5520UR
Процессор	2 x Intel Xeon E5620 (2,4 GHz; Intel-64)
Оперативная память	24 GB (6 x 4GB DDR3 1067 MHz)
Жесткий диск	4 x SATAII 500 GB (Seagate Barracuda ST3500514NS)
Сетевая карта	1 x Mellanox MT26428 InfiniBand QDR 2 x Intel Gigabit Ethernet (Intel 82575EB Gigabit Ethernet Controller)
Корпус	Rack mount 2U

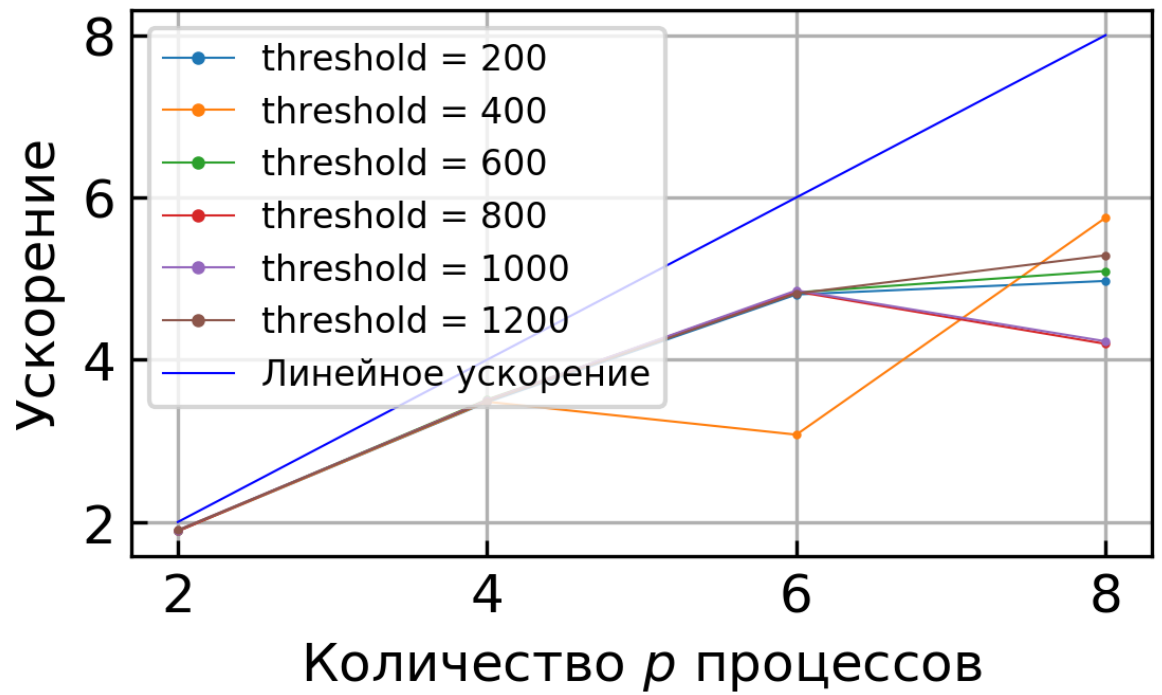
Конфигурация коммуникационной среды

Сервисная сеть	Коммутатор Gigabit Ethernet (D-Link DGS-1224T)
Вычислительная сеть	Коммутатор Infiniband QDR (Mellanox InfiniScale IV IS5030 QDR 36-Port InfiniBand Switch)

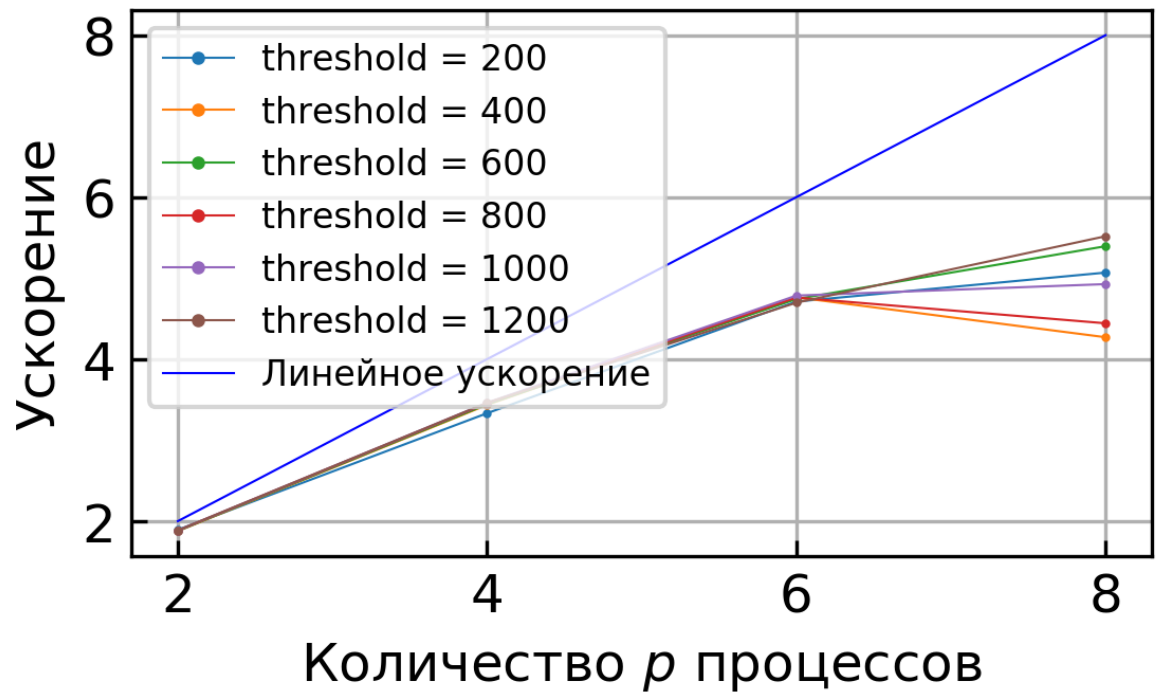
250000 элементов



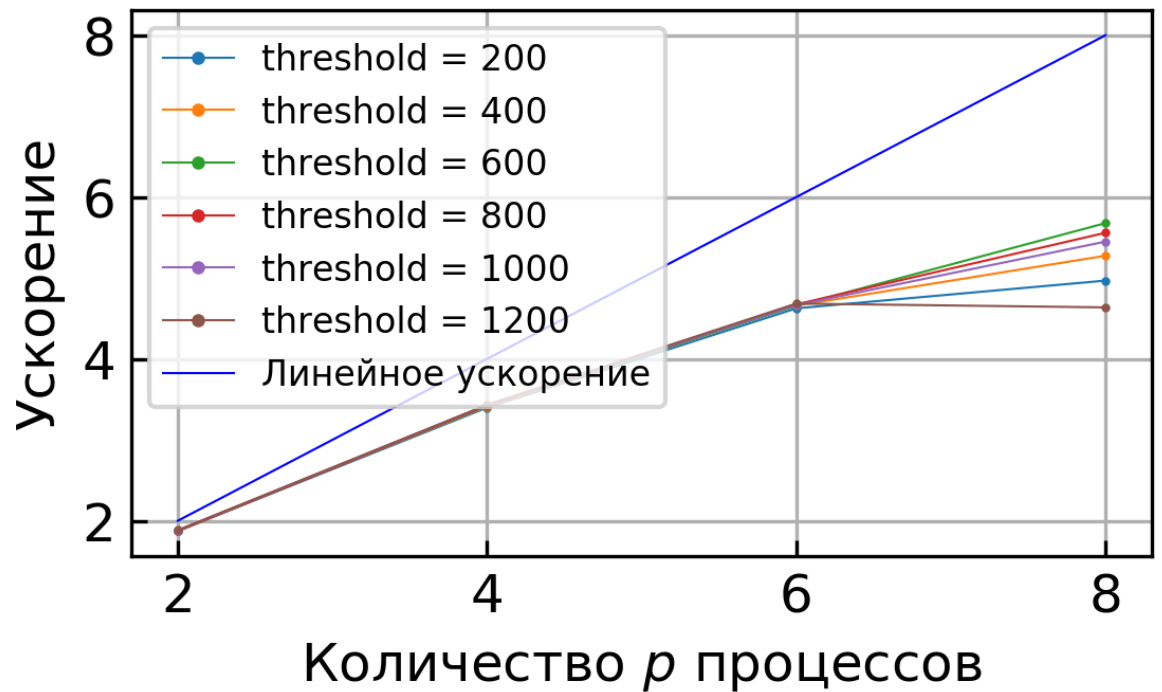
500000 элементов



750000 элементов



1000000 элементов



Заключение

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