

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Вятский государственный университет»  
Факультет автоматики и вычислительной техники  
Кафедра электронных вычислительных машин

Отчет по лабораторной работе №8 дисциплины  
«Высокопроизводительные вычислительные комплексы»

Выполнил студент группы ИВТ-42 \_\_\_\_\_/Рзаев А. Э./  
Проверил преподаватель \_\_\_\_\_/Мельцов В. Ю./

Киров 2019

## 1 Цель работы

Цель данной лабораторной работы состоит в получении практических навыков выполнения параллельных программ на кластере.

## 2 Выполнение лабораторной работы

В качестве распараллеливаемой вычислительной задачи в данной лабораторной работе используется раскраска вершин графа жадным алгоритмом.

Жадный алгоритм упорядочивает вершины  $v_1 \dots v_n$  в соответствии с некоторой перестановкой и последовательно присваивает вершине  $v_i$  наименьший доступный цвет, не использовавшийся для окраски соседей  $v_i$  среди  $v_i \dots v_{i-1}$ , либо добавляет новый. Качество полученной раскраски зависит от выбранного порядка. Для улучшения результата можно использовать несколько различных перестановок вершин.

Асимптотическая сложность данного алгоритма:  $O(kn^2 \log n)$ . Для каждой перестановки вершин (всего  $k$  перестановок), сгенерированной случайно, внешний цикл перебирает все вершины  $v_i$  и для каждой определяет цвета ее соседей, что в худшем случае будет выполнено за  $n$  шагов.

### 3 Результаты выполнение

В ходе выполнения лабораторной работы, написанная программа запускалась на различном числе ядер и для различного количества вершин графа. Результаты экспериментов представлены в таблицах 1 и 2. На рисунках 1 и 2 представлены графики зависимости времени выполнения от количества потоков, и зависимость времени выполнения от размера данных.

Таблица 1

Количество вершин графа	Количество ядер (штук)									
	1	2	4	5	8	9	16	17	23	24
1500	13966	7007	3538	2853	1822	1948	1562	1708	1817	1866
2000	25775	12931	6524	5265	3362	3558	2870	3134	3293	3374
2500	39158	19609	9943	8042	5146	5366	4387	4778	5086	5241
3000	58567	29374	14831	11972	7633	7823	6370	6947	7351	7443
3500	76614	38419	19399	15686	9996	10728	8882	9496	9981	10175

Найдем ускорение вычислений при увеличении ядер по следующей формуле:

$$K(i, i + 1) = \frac{t_{Ni+1} - t_{Ni}}{t_{Ni}},$$

где  $t_{Ni}$  – время решения при текущем количестве ядер;

$t_{Ni+1}$  – время решения при увеличенном количестве ядер;

$K(i, i + 1)$  – ускорение времени решения при текущем количестве ядер относительно времени решения при увеличенном количестве ядер.

Таблица 2

Количество вершин графа	Ускорение времени решения (в процентах)								
	K(1, 2)	K(2, 4)	K(4, 5)	K(5,8)	K(8, 9)	K(9, 16)	K(16,17)	K(17, 23)	K(23, 24)
1500	-0,4983	-0,4951	-0,1936	-0,3614	0,0692	-0,1982	0,0935	0,0638	0,0270
2000	-0,4983	-0,4955	-0,1930	-0,3614	0,0583	-0,1934	0,0920	0,0507	0,0246
2500	-0,4992	-0,4929	-0,1912	-0,3601	0,0428	-0,1824	0,0891	0,0645	0,0305
3000	-0,4985	-0,4951	-0,1928	-0,3624	0,0249	-0,1857	0,0906	0,0582	0,0125
3500	-0,4985	-0,4951	-0,1914	-0,3627	0,0732	-0,1721	0,0691	0,0511	0,0194

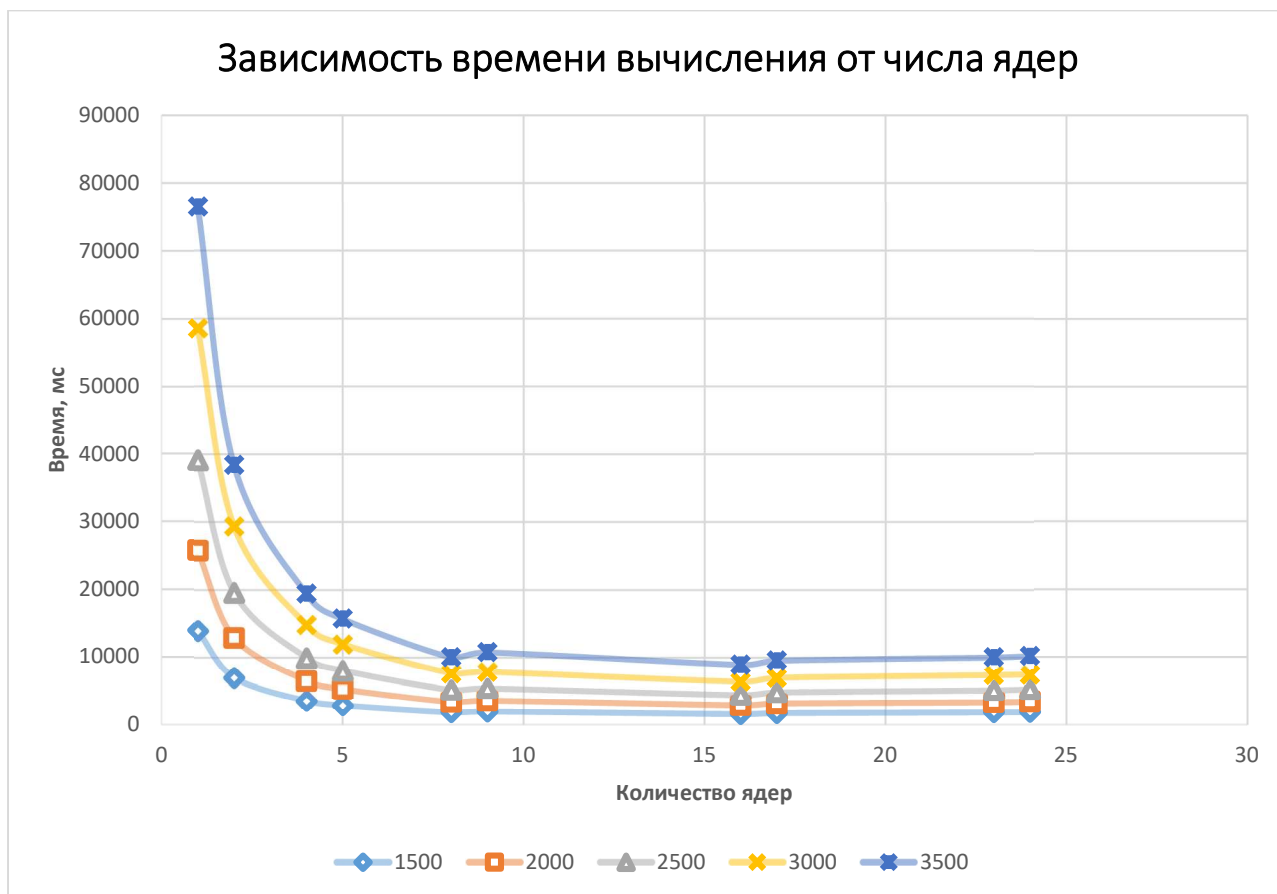


Рисунок 1 – Зависимость времени вычисления от числа ядер

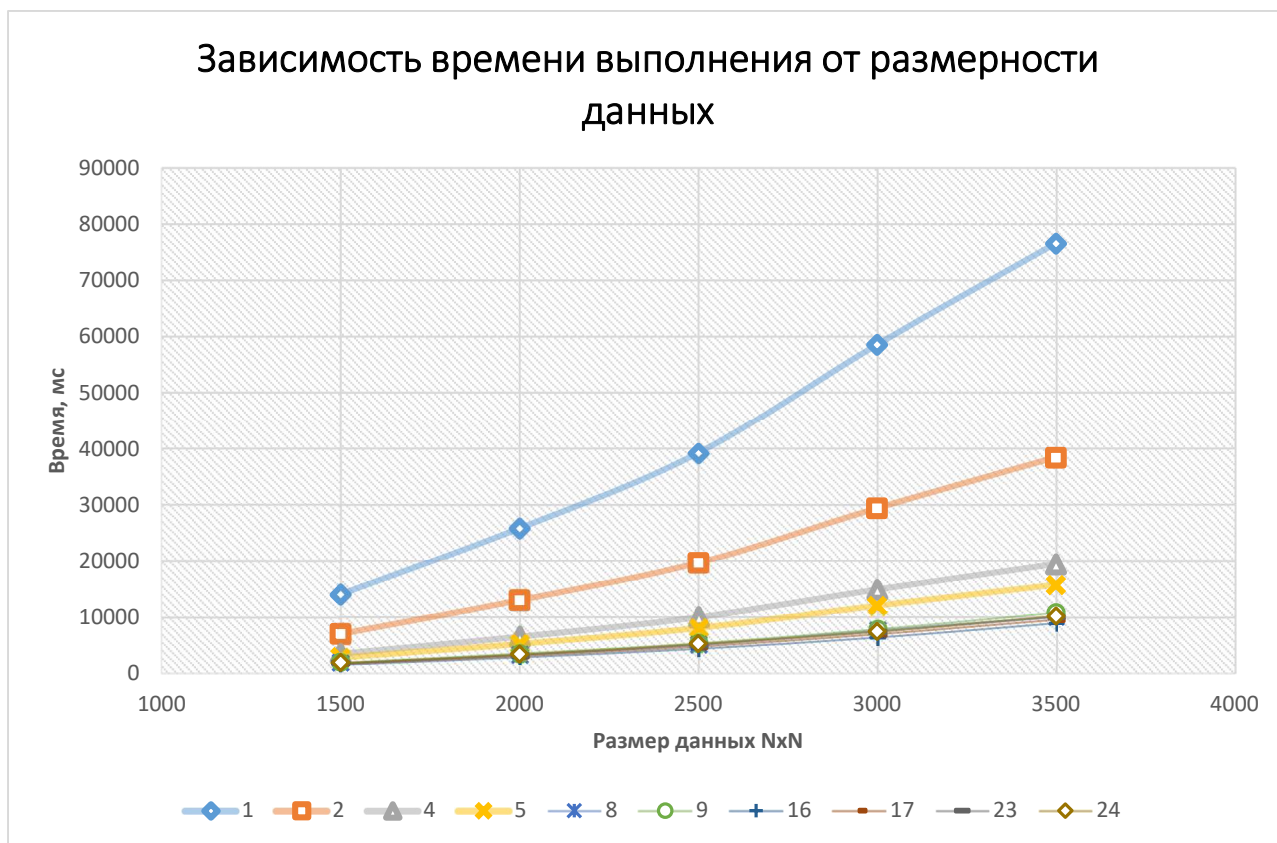


Рисунок 2 – Зависимость времени выполнения от размерности данных

## 4 Вывод

Анализируя данные в таблицах 1 и 2, можно сделать следующие выводы:

- До 8 ядер при неизменном количестве данных и увеличении числа используемых ядер время решения задачи наиболее быстро уменьшается. При количестве ядер больше 8 эффективность параллельных вычислений снижается из-за расхода времени на обмен данными между параллельными потоками.
- При решении задачи на 9 и более ядрах происходит заметное уменьшение ускорения параллельного алгоритма, при этом минимальное время решения задачи достигается на 16 ядрах, а на 17 и более ядрах наблюдается замедление параллельного алгоритма. Разница в интенсивности изменения времени связана прежде всего с тем, что при малом количестве ядер время решения задачи в большей степени состоит из времени, затрачиваемом на раскраску графа и в меньшей – на пересылку данных. В случае же большого количества ядер ситуация противоположная.
- При увеличении числа блейдов (с 1 до 2 или с 2 до 3) наблюдается увеличение времени решения задачи на 6-7%. Это можно объяснить увеличением времени на передачу данных между процессами, запущенными на разных блейдах.
- При увеличении размерности данных и одинаковом количестве используемых ядер время решения увеличивается быстрее всего при использовании небольшого числа ядер.
- При одновременном равномерном увеличении количества используемых ядер и размерности данных время решения задачи увеличивается из-за дополнительных задержек на передачу данных и организацию вычислений.

Таблица 1

	Количество ядер (штук)									
N	1	2	4	8	9	12	16	17	20	24
800	4019	2023	1050	530	590	578	530	518	514	503
1131	7937	3986	2067	1025	1140	931	809	817	807	796
1600	15755	7894	4117	2025	2151	1664	1742	1467	1419	1399
2263	31852	16010	8229	4058	4317	3274	3485	2895	2801	2755
2400	36037	18553	9397	4647	4640	3758	3149	3261	3187	3145
2771	48576	24502	12181	6122	6136	4977	4256	4321	4235	4187
3200	64145	33789	16846	8448	8465	6785	5819	5907	5752	5622
3298	68045	36279	18062	8733	9028	7231	6097	6354	6163	6002
3578	80194	40913	20383	10214	10293	8311	6977	7327	7072	6937
3919	97224	49321	24342	12430	12465	9878	8376	8720	8541	8339

Таблица 2

	Ускорение времени решения (в процентах)								
N	K(1, 2)	K(2, 4)	K(4, 8)	K(8, 9)	K(9, 12)	K(12, 16)	K(16,17)	K(17, 20)	K(20, 24)
800	-50%	-48%	-50%	11%	-2%	-8%	-2%	-1%	-2%
1131	-50%	-48%	-50%	11%	-18%	-13%	1%	-1%	-1%
1600	-50%	-48%	-51%	6%	-23%	5%	-16%	-3%	-1%
2263	-50%	-49%	-51%	6%	-24%	6%	-17%	-3%	-2%
2400	-49%	-49%	-51%	0%	-19%	-16%	4%	-2%	-1%
2771	-50%	-50%	-50%	0%	-19%	-14%	2%	-2%	-1%
3200	-47%	-50%	-50%	0%	-20%	-14%	2%	-3%	-2%
3298	-47%	-50%	-52%	3%	-20%	-16%	4%	-3%	-3%
3578	-49%	-50%	-50%	1%	-19%	-16%	5%	-3%	-2%
3919	-49%	-51%	-49%	0%	-21%	-15%	4%	-2%	-2%

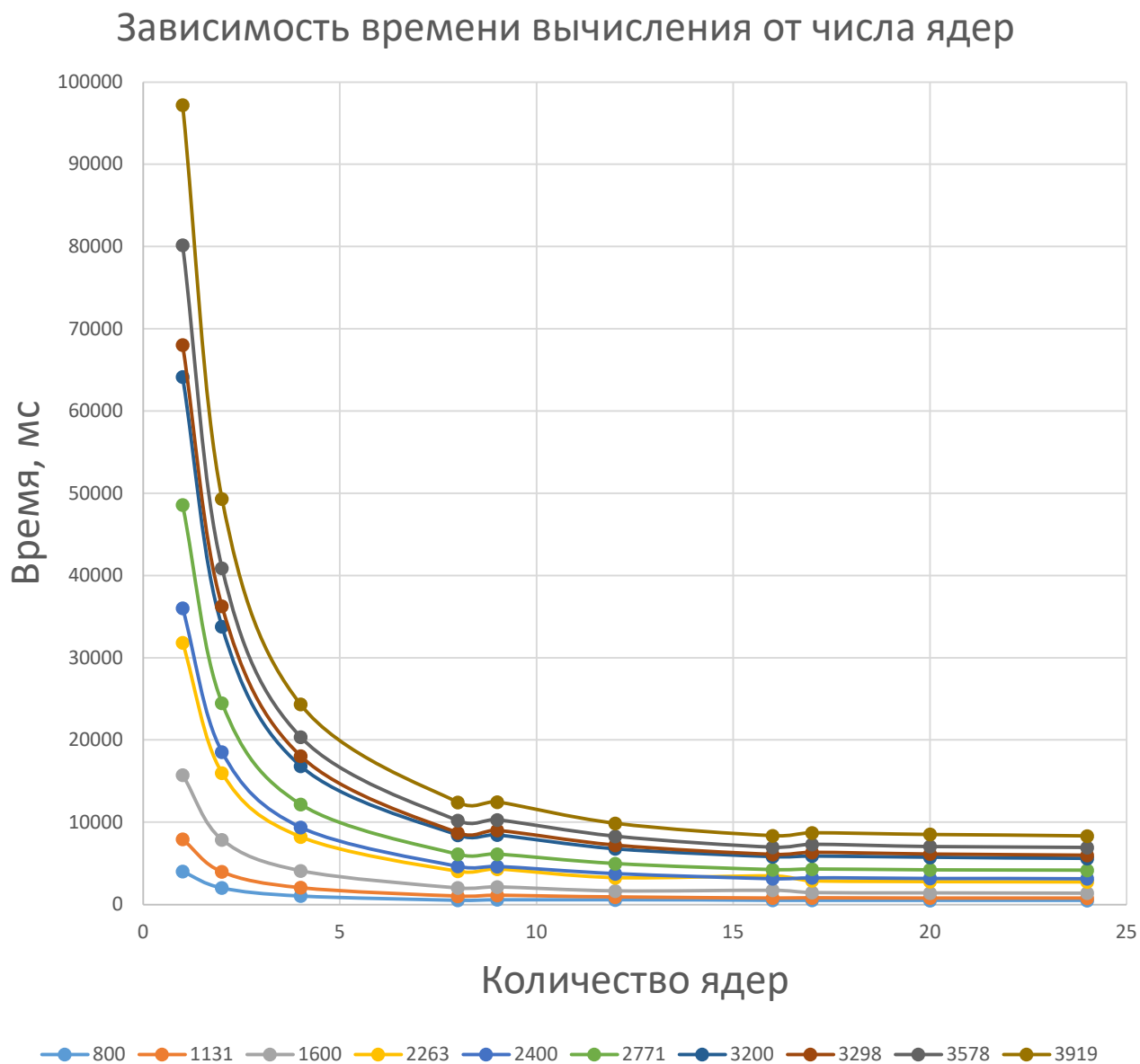


Рисунок 1 – Зависимость времени вычисления от числа ядер

## Зависимость времени выполнения от размерности данных

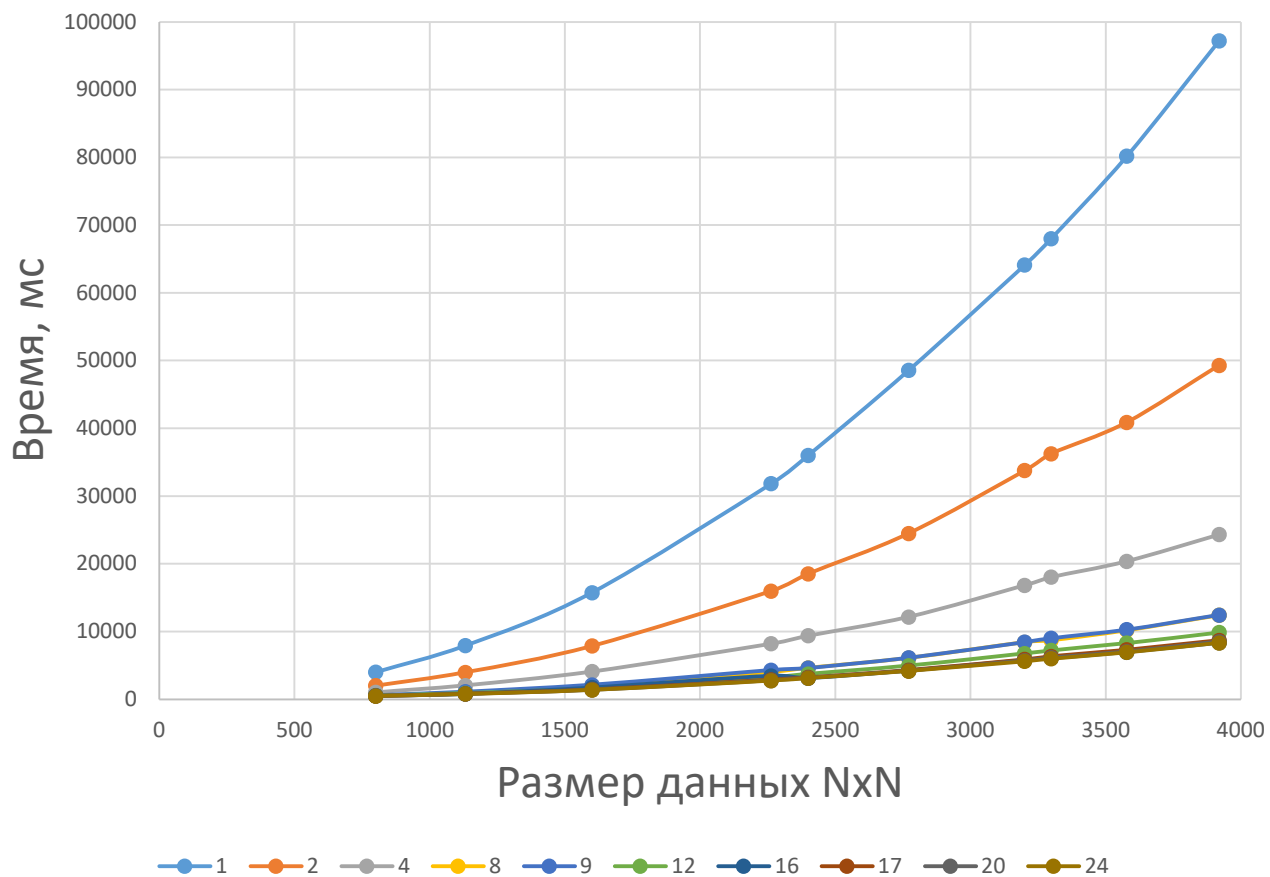


Рисунок 2 – Зависимость времени выполнения от размерности данных