

Задание на защиту лабораторной работы №6

Выполнил Титов Дмитрий, Б22-505

1. Данные и графики

num of proc ros num	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0,16 434 2															
2	0,17 727 9	1,00 049 4														
3	0,18 127 6	0,91 771 8	0,97 061 3													
4	0,18 168 6	0,88 487 4	0,93 058 4	0,96 894 1												
5	0,16 614 3	0,86 293 6	0,89 717 2	0,93 011	0,96 277 6											
6	0,20 295 4	0,90 098 1	0,93 836 2	0,97 098 7	1,00 058 6	1,03 041										
7	0,17 537 2	0,89 173 1	0,91 776 7	0,94 230 4	0,96 680 6	0,99 145 6	1,01 769 4									
8	0,17 496 5	0,89 429 1	0,91 594 3	0,93 628 5	0,95 741 8	0,97 835 5	1,00 053 3	1,02 284 9								
9	0,20 099 5	0,85 772	0,87 907 5	0,89 760 9	0,92 216 5	0,94 158 9	0,96 288 1	0,99 177 5	1,01 365 5							
10	0,20 800 4	1,05 324 7	1,07 123 7	1,08 876 9	1,10 750 5	1,13 012 9	1,14 932 5	1,16 867 2	1,19 556 1	1,22 610 4						
11	0,22 153 4	0,80 633 5	0,82 784 4	0,84 359 1	0,86 347 7	0,87 960 8	0,89 684 4	0,91 419 7	0,93 774 3	0,96 674 9	0,99 598 4					
12	0,23 9	0,83 055 2	0,85 059 2	0,86 897 3	0,88 393 7	0,90 366 9	0,92 428 9	0,94 188 6	0,96 306 1	0,98 217 3	1,00 962 4	1,04 01				
13	0,26 986 3	1,00 234 7	1,02 094 1	1,03 835 8	1,05 675 5	1,07 567 8	1,09 533	1,11 095	1,13 137 8	1,15 536 6	1,17 820 2	1,20 498 2	1,24 529 6			

14	0,28 032 3	1,00 590 7	1,02 370 6	1,03 993 4	1,05 677 4	1,07 288 6	1,08 994 7	1,10 428 9	1,12 230 8	1,14 3611	1,16 160 6	1,18 534 7	1,21 650 3	1,25 688 4		
15	0,30 142 4	0,97 296 5	0,98 937 5	1,00 426	1,01 938 6	1,03 500 9	1,05 1168	1,06 818 1	1,08 530 6	1,10 501 3	1,12 734 2	1,15 158 8	1,18 403 4	1,21 472 7	1,24 967	
16	0,04 361 2	1,00 046 6	1,00 063 8	0,99 955 1	0,99 492 9	0,99 908 8	0,99 922 7	0,99 969 5	0,99 921 6	0,99 991 9	0,99 999	0,99 666 2	0,99 698 3	0,99 999	0,99 755 3	0,99 841

Таблица 1. Время задержки каждого процессора от их количества

Время задержки процессора на разном числе процессоров

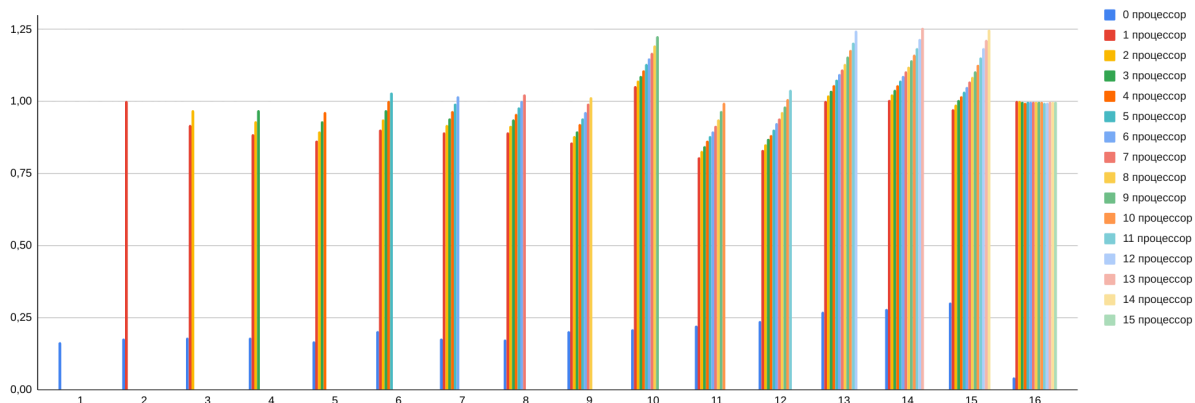


Рис. 1. Гистограмма времени задержек процессоров от их числа

2. Анализ данных

1. Задержка начала работы для 0-го процессора на 16 процессорах

- Для 0-го процессора на 16 процессорах время начала работы значительно меньше, чем для других количеств процессоров. Это объясняется особенностями распределения данных в MPI и синхронизации. На 16 процессорах, когда число задач значительно больше, нагрузка на каждый процессор становится более сбалансированной, и данные передаются более эффективно, что снижает время начала работы для 0-го процессора. Однако при меньшем количестве процессоров, нагрузка на процессоры может быть менее сбалансирована, и процессор 0 может ждать, пока данные будут отправлены другим процессорам.

2. Задержки для некоторых процессоров на определенном числе процессоров

- Например, для 3-го процессора на 4 процессорах время задержки больше, чем для 3-го процессора на 11 процессорах. Это может быть связано с несколькими факторами:

- **Балансировка данных:** Когда на программе запускается больше процессоров, каждый процесс получает меньше данных, что может улучшить синхронизацию и уменьшить время, которое процессор тратит на ожидание других процессов.
- **Конкуренция за ресурсы:** При меньшем числе процессоров конкуренция за ресурсы (например, за память или каналы передачи данных) может быть выше, что замедляет выполнение.
- **Коммуникационные задержки:** На меньшем числе процессоров (например, 4) коммуникативные задержки могут оказывать большее влияние, так как данные передаются между процессами чаще, и это может вызвать задержки при запуске каждого процесса. Когда процессоров больше (например, 11), данные могут быть распределены более эффективно, и каждый процесс получает свою часть быстрее.

3. Увеличение задержек при увеличении числа процессоров (от 1 до 16)

- На первых этапах (например, от 1 до 4 процессоров) время задержки растет, потому что каждый процесс должен ожидать получения данных от других процессов. На небольшом числе процессоров нагрузка не всегда сбалансирована, и время передачи данных между процессами может занимать значительное время.
- Когда процессоров становится больше (например, 16), распределение данных становится более равномерным, и для процессоров может быть быстрее начинать работать, так как каждый из них обрабатывает меньший объем данных и имеет меньше ожидания.

4. Проблемы с балансировкой на малом количестве процессоров

- На 4 процессорах процессор 3 имеет большую задержку, вероятно, из-за неэффективного распределения данных, когда количество данных, которое нужно обработать этим процессом, не оправдывает его ожидания из-за недостаточной параллельности на малом числе процессоров.
- С увеличением числа процессоров балансировка улучшается, и процессор 3 получает меньшую нагрузку, что уменьшает время задержки.

5. Влияние количества процессоров на время старта для каждого из них

- На более высоких числах процессоров (например, 11, 12 и 16) время начала работы для каждого процессора может быть более сбалансированным, так как нагрузка делится между большим числом процессоров. В таких случаях распределение данных происходит быстрее, и каждый процесс может начать выполнение раньше.

Выводы:

- На начальных этапах (при небольшом количестве процессоров) процессы могут иметь большие задержки из-за несбалансированного распределения работы или коммуникативных задержек.
- На большем числе процессоров время задержки начинает уменьшаться, так как данные эффективно распределяются, и количество операций на каждом процессе становится более равномерным.
- Задержка для процессора 0 при 16 процессорах минимальна, вероятно, потому что программа при таком числе процессоров эффективно балансирует нагрузку и минимизирует время ожидания для первого процесса.
- Поведение задержки для процессоров 3, 4 и других зависит от специфики распределения данных и нагрузочного баланса, что может изменяться в зависимости от числа процессоров