

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ М. В. ЛОМОНОСОВА
ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ И КИБЕРНЕТИКИ

Задача №2 по практикуму

Отчёт О выполненном задании

Выполнил:
студент 423 группы
Латыпов Ш. И.

Москва
2023

Содержание

Описание задачи	2
Алгоритм программы	2
Работа программы	3
Выводы	6

Описание задачи

Задача состоит в том, что необходимо исследовать проблему масштабируемости при программировании с использованием технологии MPI.

В данном варианте рассматривается проблема синхронизации нитей из-за использования функций обмена данными с блокировкой. Для явной демонстрации этой проблемы написаны две программы, в одной используется оптимальный алгоритм решения задачи, в другой используются функции обмена данными с блокировкой. Это не меняет логику программы, но замедляет её работу вследствие задержки времени на синхронизацию нитей.

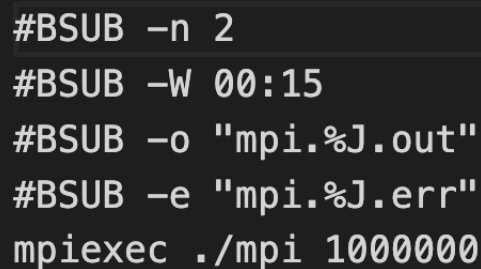
Алгоритм программы

Для демонстрации этой проблемы масштабируемости была написана программа, которая в параллельном суммирует ранги всех процессов через буферы.

Процессы передают друг другу свои ранги в виде буферов, суммируют их в общее значение и передают дальше. Для этого используются пересылки MPI_Send и MPI_Recv в обычной программе и MPI_Isend и MPI_Irecv в оптимизированной соответственно. В программе вызывается 1000 пересылок буферов.

Тестирование программы выполнялось на параллельной вычислительной системе *Polus*: 3 вычислительных узла, 2 десятиядерных процессора IBM POWER8. Использовался компилятор `mpicc`.

Запуски проводились через планировщик IBM Spectrum LSF. Файл конфигурации имеет вид, представленный ниже на фотографии. Программа запускалась для размера буферов 10^6 , $5 \cdot 10^6$ и 10^7 с использованием 2, 4, 8, 16 нитей.



```
#BSUB -n 2
#BSUB -W 00:15
#BSUB -o "mpi.%J.out"
#BSUB -e "mpi.%J.err"
mpirun ./mpi 1000000
```

Рис. 1: Файл конфигурации LSF

Работа программы

1. Размер буфера 10^6

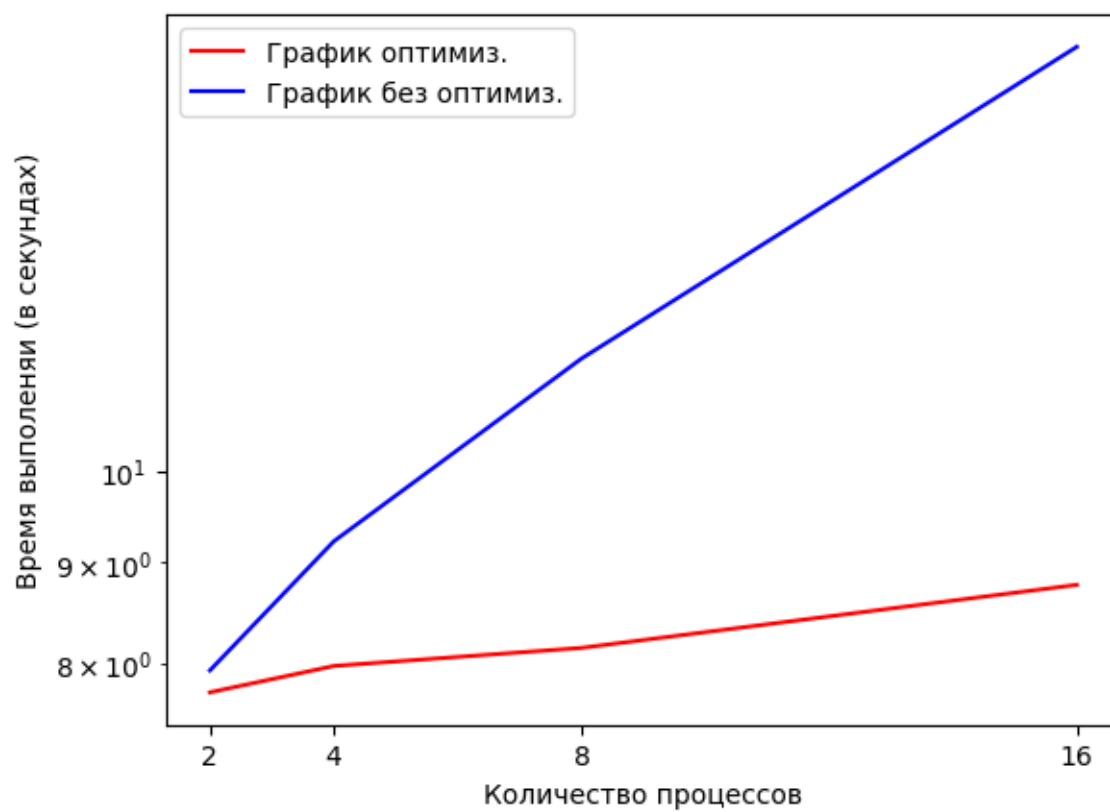


Рис. 2: График времени работы

1000000	2	4	8	16
Оптимизированные	7,731160	7,972135	8,141161	8,760458
Без оптимизации	7,930942	9,214684	11,395896	16,374117
Ускорение с оптим.	1	0,9697729	0,94963857	0,88250642
Ускорение без оптим.	1	0,86068517	0,69594721	0,48435845
Разница (%)	2,58411416	15,5861608	39,9787659	86,9093823

Рис. 3: Таблица результатов

2. Размер буфера 5×10^6

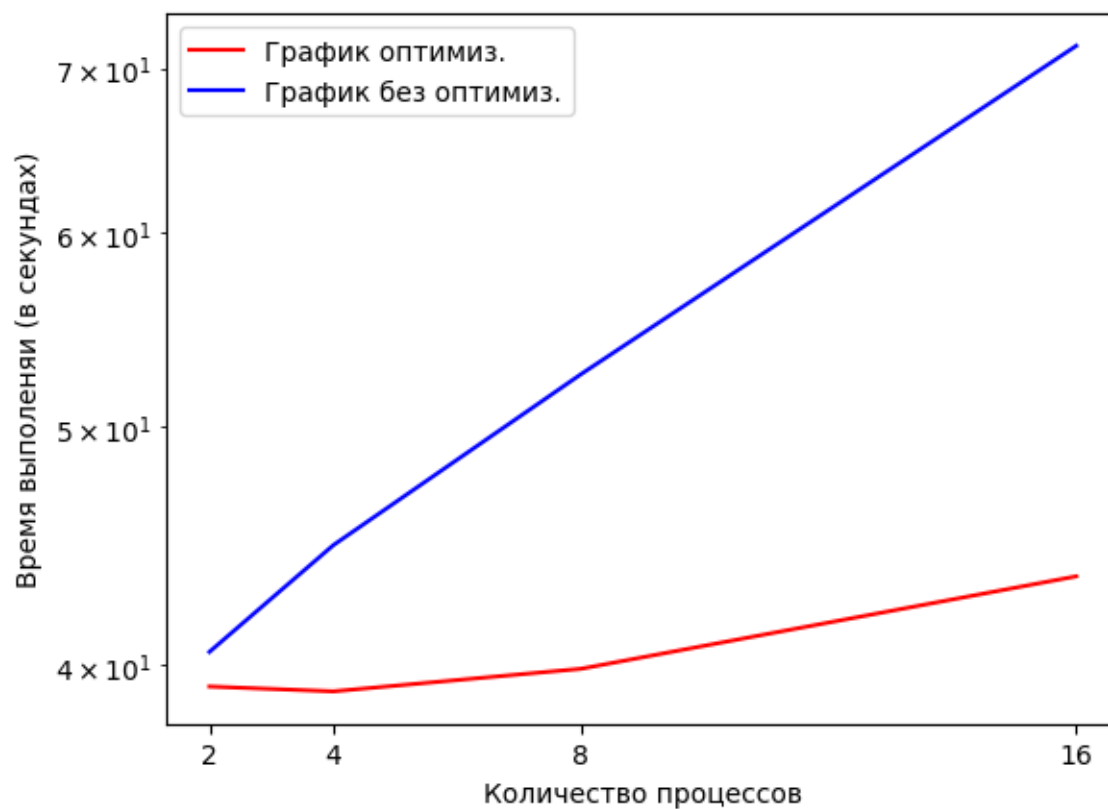


Рис. 4: График времени работы

5000000	2	4	8	16
Оптимизированные	39,187795	39,010939	39,845269	43,455802
Без оптимизации	40,482044	44,742446	52,526260	71,479812
Ускорение с оптим.	1	1,004533511	0,983499333	0,901785116
Ускорение без оптим.	1	0,904779412	0,770701062	0,56634234
Разница (%)	3,30268391	14,69205233	31,82558802	64,4885367

Рис. 5: Таблица результатов

3. Размер буфера 10^7

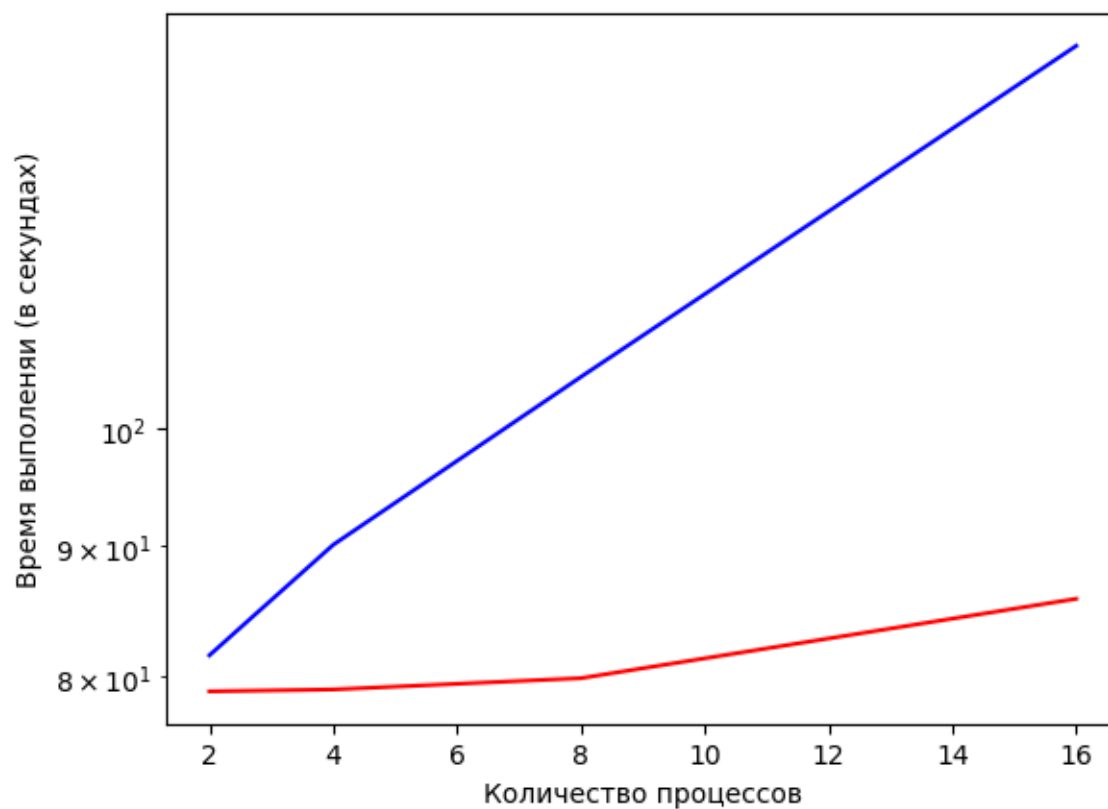


Рис. 6: График времени работы

10000000	2	4	8	16
Оптимизированные	78,906769	79,040350	79,846698	85,760354
Без оптимизации	81,510939	90,033780	104,715911	141,058856
Ускорение с оптим.	1	0,998309964	0,988228336	0,920084465
Ускорение без оптим.	1	0,905337302	0,778400706	0,577850563
Разница (%)	3,30031255	13,90862958	31,14620176	64,4802636

Рис. 7: Таблица результатов

Выводы

Программа практически не ускоряется при 2 нитях, разница по времени не превышает 4%, но уже при 4 нитях и более наблюдается значительное ускорение программы, вплоть до 65% эффективности