

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**
**НОВОСИБИРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**
Факультет информационных технологий
Кафедра параллельных вычислений

ОТЧЕТ

О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ

«Умножение матрицы на матрицу в MPI 2D решетка»

студента 2 курса, 19202 группы

Бакшеева Ивана Дмитриевича

Направление 09.03.01 – «Информатика и вычислительная техника»

Преподаватель:

к. т. н., доцент

А. Ю. Власенко

Новосибирск 2021

СОДЕРЖАНИЕ

ЦЕЛЬ.....	3
ЗАДАНИЕ	3
ОПИСАНИЕ РАБОТЫ	4
ЗАКЛЮЧЕНИЕ	5
Приложение 1. Графики по таблице	6
Приложение 2. Профилирование (скриншот).....	6

ЦЕЛЬ

Изучить топологии процессов MPI, изучить метод перемножения матриц на 2D решетках процессов.

ЗАДАНИЕ

1. Реализовать параллельный алгоритм умножения матрицы на матрицу при 2D решетке.
2. Исследовать производительность параллельной программы во зависимости от размера матрицы и размера решетки.
3. Выполнить профилирование программы с помощью MPE при использовании 16-ти ядер.
4. Сделать выводы, найти самую эффективную решетку.

ОПИСАНИЕ РАБОТЫ

1. Реализовал параллельный алгоритм умножения матрицы на матрицу (из задания). При создании решетки не использовал перенумерацию процессов. Корректность программы проверял, повторно считая произведение, но уже последовательным алгоритмом, а потом сравнивая результаты. В качестве начальных значений для матриц брал случайные целые числа.
2. Запустил на кластере программу с различным числом процессов, с различными по форме решетками.
3. Провел профилирование программы на 16 процессах с решеткой 4x4.
4. Сделал выводы по результатам измерений.

ЗАКЛЮЧЕНИЕ

Таблица по зависимости времени выполнения от числа процессов:

Число процессов	P1	P2	N1	N2	N3	T (с.)	Sp	Ep
1	1	1	2400	2400	2400	65,53	1,00	100%
2	1	2	2400	2400	2400	29,78	2,20	110%
4	2	2	2400	2400	2400	31,67	2,07	52%
8	2	4	2400	2400	2400	5,21	12,58	157%
12	2	6	2400	2400	2400	10,65	6,15	51%
16	4	4	2400	2400	2400	6,05	10,83	68%
24	4	6	2400	2400	2400	8,51	7,70	32%

Где P1, P2 – размеры решетки, N1-N3 – размеры матриц, T – время перемножения, Sp – ускорение, Ep – эффективность.

По результатам таблицы можно сделать вывод, что решетка 2 на 4 обладает большой эффективностью. Эффективность может быть более 100%, так как уменьшение размера матрицы (ведь в алгоритме используется перемножение подматриц) уменьшает время выполнения очень сильно, так как обычный алгоритм перемножения матриц выполняется за $O(N^3)$.

Таблица зависимости времени выполнения от формы решетки:

Число процессов	P1	P2	N1	N2	N3	T (с.)
24	1	24	2400	2400	2400	3,25
24	2	12	2400	2400	2400	4,1
24	3	8	2400	2400	2400	6,01
24	4	6	2400	2400	2400	8,53
24	6	4	2400	2400	2400	4,57
24	8	3	2400	2400	2400	8,32
24	12	2	2400	2400	2400	8,75
24	24	1	2400	2400	2400	9,54

Тут замечен немного парадоксальный результат, что приближение формы решетки к квадрату не уменьшает время выполнения, а увеличивает его. Также видно, как передача столбцов замедляет программу (1x24 против 24x1). Самой эффективной оказалась решетка 1x24 процесса.

Таблица зависимости времени выполнения от размеров матриц:

Число процессов	P1	P2	N1	N2	N3	T (с.)
16	4	4	800	800	320	0,06
16	4	4	1600	1600	640	0,93
16	4	4	2400	2400	960	3,97
16	4	4	3200	3200	1280	9,64
16	4	4	4000	4000	1600	19,86

С ростом размера матрицы время выполнения значительно увеличивается. По результатам профилирования можно заметить, что наибольшее время занимает неэффективный наивный алгоритм умножения подматриц (красный цвет). Раздача данных по процессам занимает минимальное время (синий цвет).

Приложение 1. Графики по таблице



Приложение 2. Профилирование (скриншот)

