



**Спецкурс: системы и средства параллельного
программирования**

Отчёт № 5

**Параллельный алгоритм DNS матричного умножения.
Разработка параллельной MPI программы и исследование ее
эффективности**

**Работу выполнила
Домрачева Д. А.**

Постановка задачи и формат данных

Задача:

Разработать параллельную программу с использованием технологии MPI, реализующую алгоритм матричного умножения $A \times B = C$. Тип данных – double. Провести исследование эффективности разработанной программы на системе Blue Gene/P.

Формат командной строки:

<имя файла матрицы A> <имя файла матрицы B> <имя файла матрицы C> <размер одного измерения решетки>

Формат файла с матрицей:

Матрица представляются в виде бинарного файла следующего формата:

Тип	Значение	Описание
Число типа int	N – натуральное число	Число строк матрицы
Число типа int	M – натуральное число	Число столбцов матрицы
Массив чисел типа double	$N \times M$ элементов	Массив элементов матрицы

Элементы матрицы хранятся построчно.

Описание алгоритма

Математическая постановка:

Алгоритм матричного умножения можно представить в следующем виде: $C_{ij} = \sum_{k=1}^N A_{ik} \cdot B_{kj}$ для каждого элемента матрицы.

Результаты выполнения

Суммарное время работы процессов:

N	M	mapping	8	64	125
1024	1024	standard	24.312	34.7795	43.65
2048	2048	standard	190.855	233.836	247.444
4096	4096	standard	1562.65	1616.68	1851.85
1024	1024	random	–	–	43.4642
2048	2048	random	–	–	254.629
4096	4096	random	–	–	1809.74

Максимальное время:

N	M	mapping	8	64	125
1024	1024	standard	3.064204	0.56	0.364211
2048	2048	standard	23.929773	3.714298	2.037828
4096	4096	standard	195.646361	25.479585	15.037234

1024	1024	random	–	–	0.363577
2048	2048	random	–	–	2.099520
4096	4096	random	–	–	14.723077

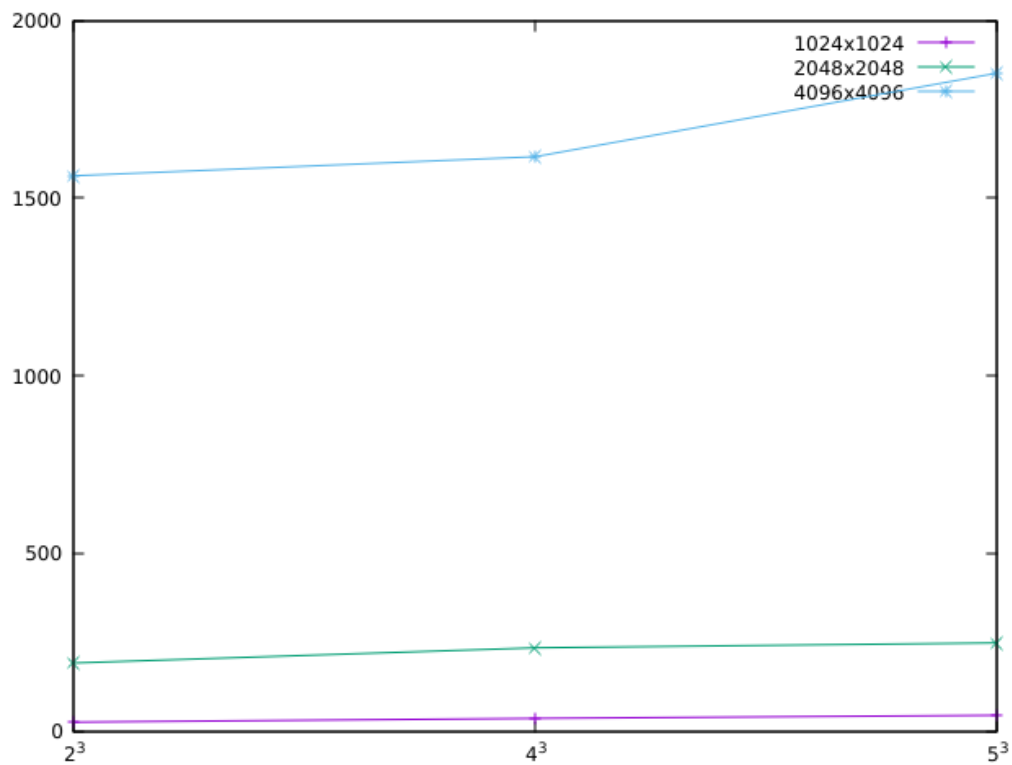
Ускорение $S(n)=T_1(n)/T_p(n)$:

N	M	mapping	8	64	125
1024	1024	standard	8.22128	44.9821	69.1652
2048	2048	standard	8.87098	57.1567	104.212
4096	4096	standard	9.00555	69.1511	117.1694

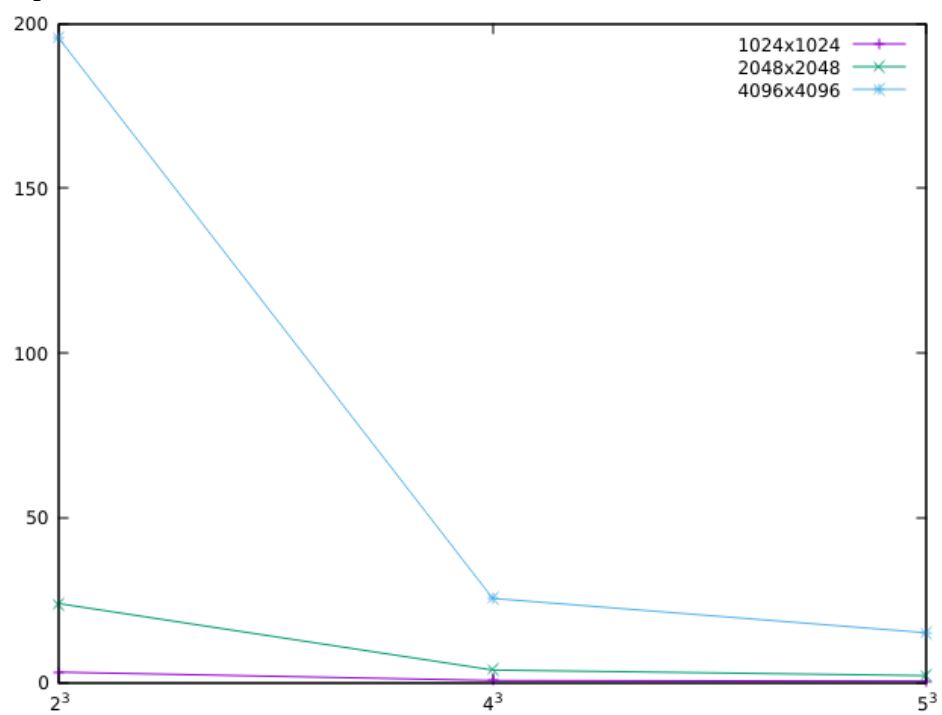
Эффективность $E(n)=T_1(n)/(p \cdot T_p(n))$:

N	M	mapping	8	64	125
1024	1024	standard	1.0276	0.7028	0.5533
2048	2048	standard	1.1089	0.8931	0.8337
4096	4096	standard	1.1256	1.0804	0.9373

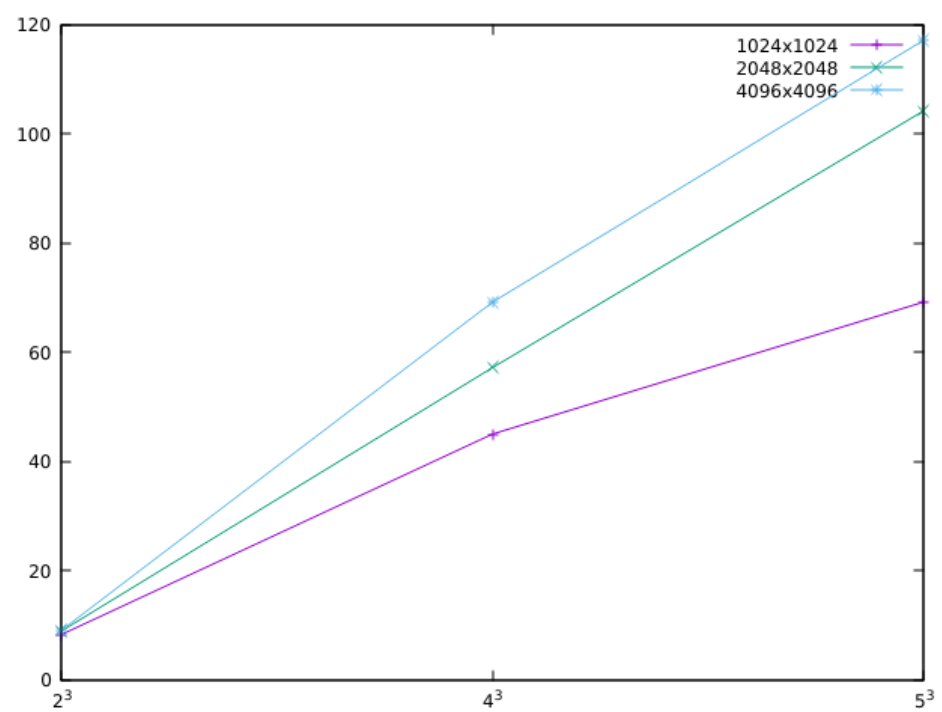
Суммарное время работы процессов:



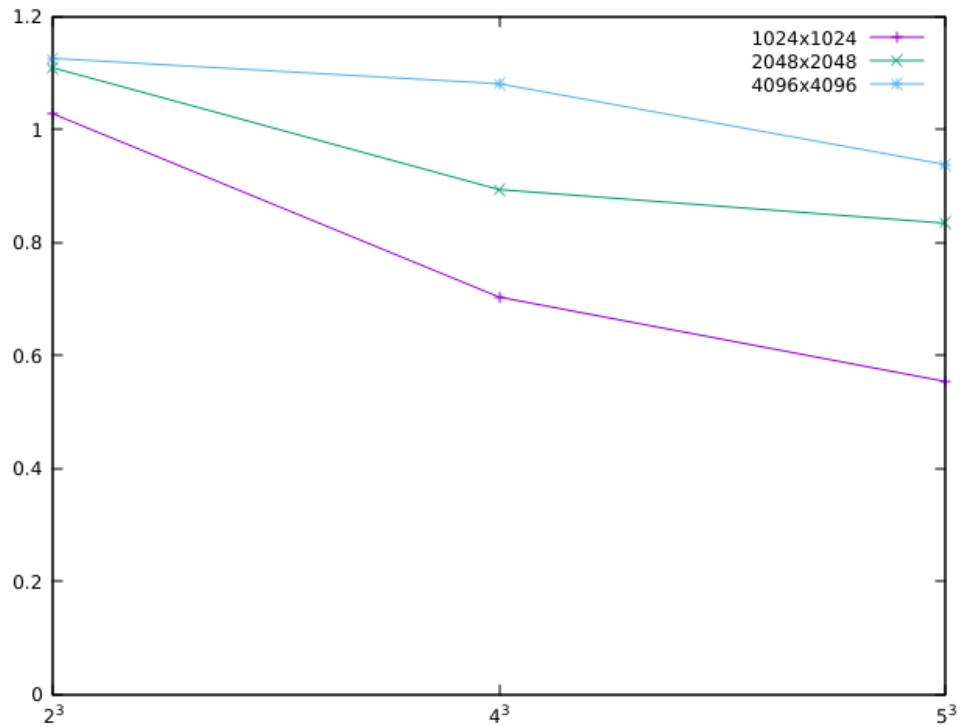
Максимальное время:



Ускорение:



Эффективность:



Выводы

С ростом числа процессов наблюдается рост ускорения, близкий к линейному. Снижение эффективности можно объяснить тем, что накладных расходов на пересылку данных становится больше, чем непосредственно вычислений. Влияния мэппинга на время работы программы выявлено не было.