



## **Системы и средства параллельного программирования**

### **Отчёт**

## **Параллельный алгоритм умножения матрицы на вектор**

Работу выполнил  
Евгений Кислов, 323

Москва 2020

## Постановка задачи и формат данных

### Задача

Разработать параллельную программу с использованием технологии MPI, реализующую алгоритм умножения плотной матрицы на вектор  $Ab=c$ . Тип данных – double. Провести исследование эффективности разработанной программы на системе Blue Gene/P.

### Требуется:

1. Разработать параллельную программу с использованием технологии MPI. Предусмотреть равномерное распределение элементов матрицы блоками строк или столбцов, в зависимости от соотношения  $m$  и  $n$ . Вектора  $b$  и  $c$  распределены по процессам равномерно.
2. Исследовать эффективность разработанной программы в зависимости от размеров матрицы и количества используемых процессов. Построить графики времени работы, ускорения и эффективности разработанной программы. Время на ввод/вывод данных не включать.
3. Исследовать влияние мэппинга параллельной программы на время работы программы.
4. Построить таблицы: времени, ускорения, эффективности.

m	n	мэппинг	32	64	128	256	512

Для 512 процессоров рассмотреть два варианта мэппинга – стандартный, принятый по умолчанию и произвольный. Для произвольного мэппинга предусмотреть генерацию строк файла для задания случайного значения XYZT.

Ускорение (speedup), получаемое при использовании параллельного алгоритма для  $p$  процессоров, определяется величиной:

$$\text{Speedup}(n) = T1(n)/Tp(n),$$

где  $T1(n)$ - время выполнения задачи на одном процессоре.

$Tp(n)$ - время параллельного выполнения задачи при использовании  $p$  процессоров.

5. Построить графики – для каждого из заданных значений размеров матрицы (512x512, 1024x1024, 2048x2048, 4096x4096, 4096x1024, 1024x4096).

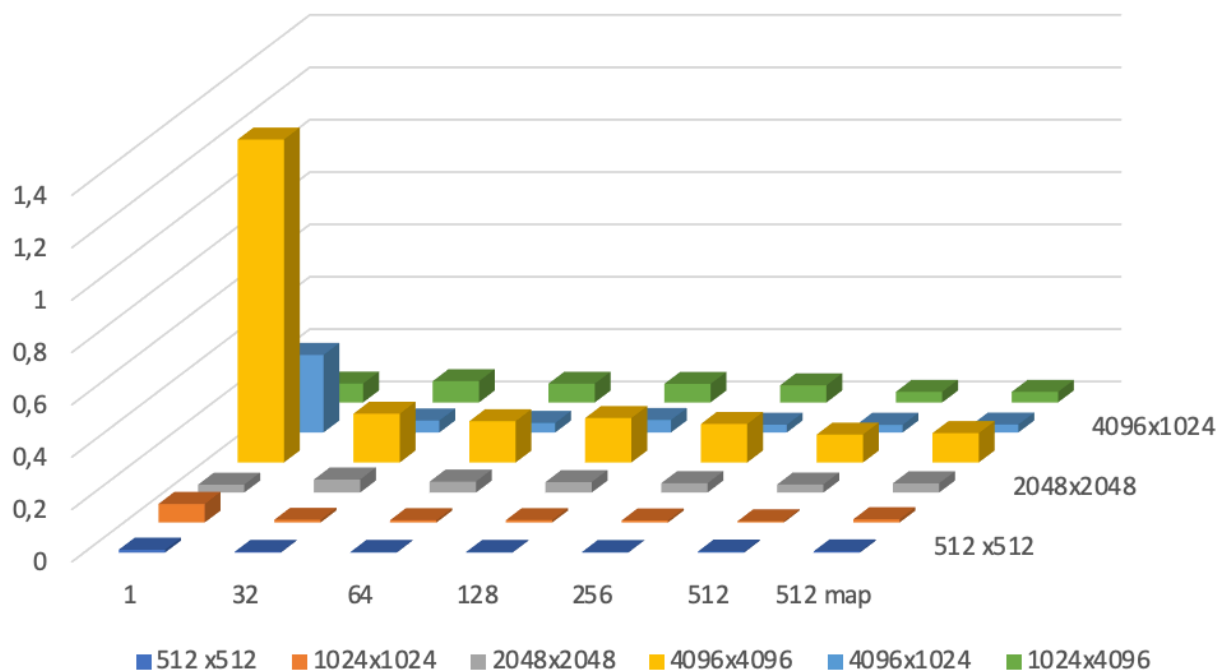
### Формат командной строки

- имя файла – матрица  $A$  размером  $m \times n$
- имя файла - вектор  $b$
- имя файла – результат, вектор  $c$

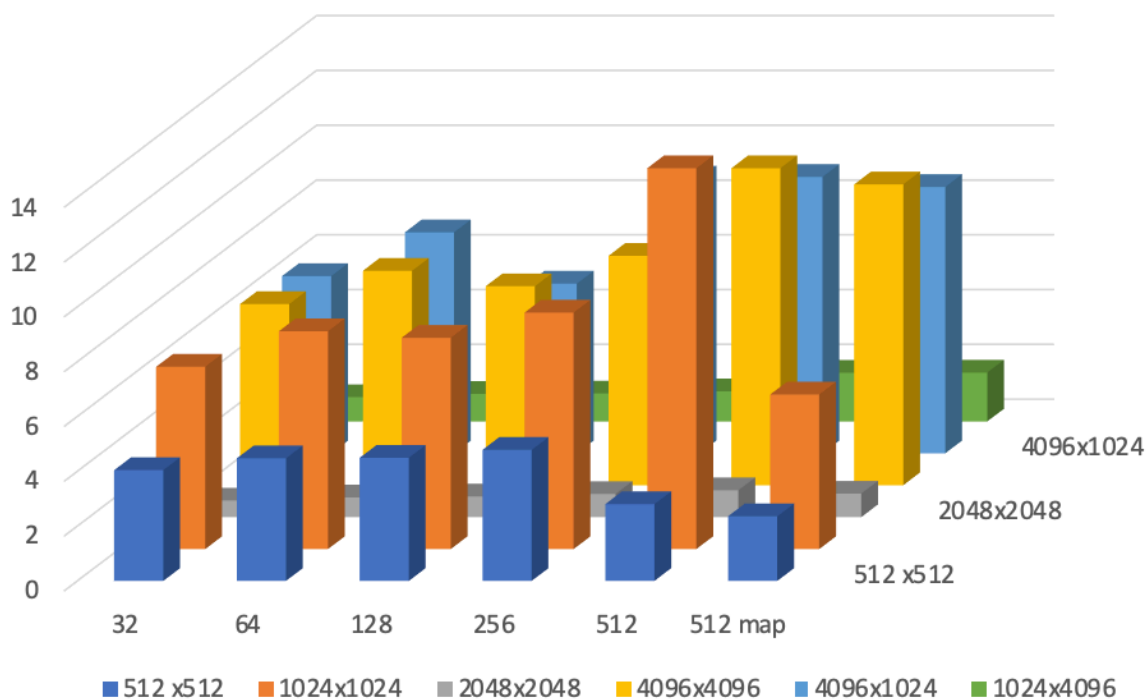
Формат задания матрицы  $A$  – как в первом задании.

# Результаты

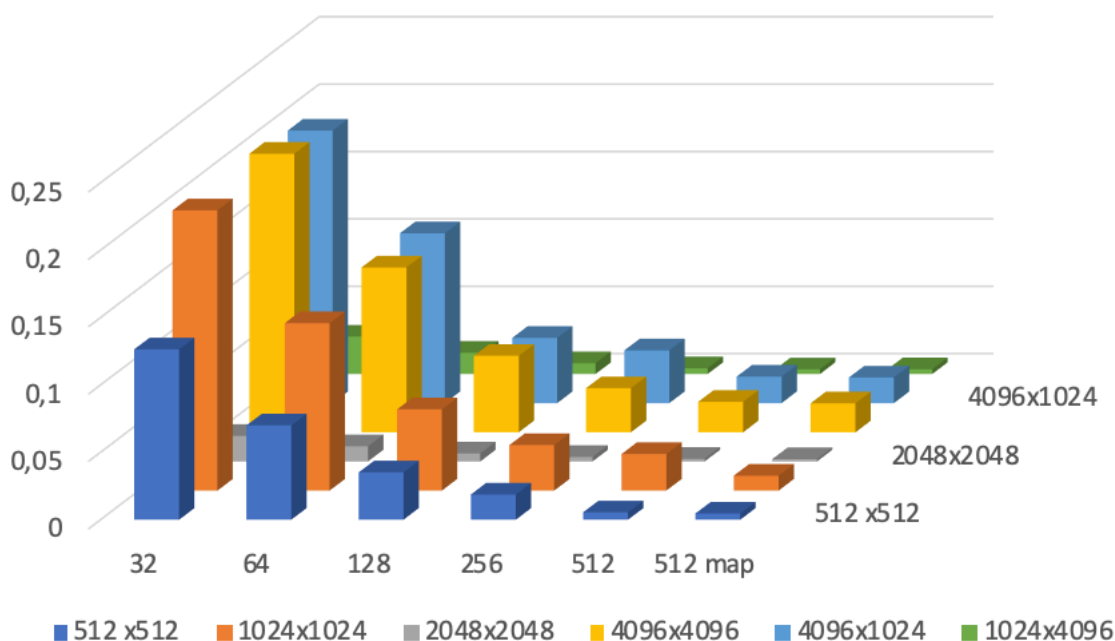
## Время выполнения



## Ускорение



## Эффективность



## Таблицы

	A	B	C	D	E	F	G
1	Время	512 x512	1024x1024	2048x2048	4096x4096	4096x1024	1024x4096
2	1	0,0110203	0,07124313	0,03012302	1,23384021	0,29631032	0,0723901
3	32	0,00272534	0,01070898	0,04972847	0,18656258	0,04578827	0,08152663
4	64	0,00246134	0,00896303	0,04176783	0,15762873	0,03672624	0,07173687
5	128	0,00245158	0,00924017	0,03978272	0,16987391	0,04782672	0,0708392
6	256	0,00230368	0,00825485	0,03565123	0,14736824	0,02954832	0,06572372
7	512	0,00392944	0,00512552	0,03066261	0,10662342	0,02936873	0,04076872
8	512 map	0,00467121	0,01263767	0,03516271	0,11236101	0,03047101	0,0405611
9							
10							
11	Ускорение	512 x512	1024x1024	2048x2048	4096x4096	4096x1024	1024x4096
12	1	1	1	1	1	1	1
13	32	4,04364226114907	6,6526531938616	0,605749985873283	6,61354602836217	6,47131503330438	0,887931955484975
14	64	4,47735786197762	7,94855422775557	0,721201460549902	7,82750841169627	8,06808211240791	1,0091059172222
15	128	4,49518269850464	7,71015360107011	0,757188548193789	7,26327079891197	6,19549741232516	1,02189324554766
16	256	4,7837807334352	8,63045724634609	0,844936345814717	8,37249742549684	10,0279921159646	1,10143035117306
17	512	2,8045472128344	13,8996882267555	0,982402346049472	11,5719436686612	10,0893133615243	1,77562847202463
18	512 map	2,35919601131184	5,63736274170793	0,856675153877503	10,9810352363333	9,72433536006847	1,78471737699421
19							
20							
21	Эффективность	512 x512	1024x1024	2048x2048	4096x4096	4096x1024	1024x4096
22	1	1	1	1	1	1	1
23	32	0,126363820660908	0,207895412308175	0,0189296870585401	0,206673313386318	0,202228594790762	0,0277478736089055
24	64	0,0699587165934003	0,124196159808681	0,0112687728210922	0,122304818932754	0,126063783006374	0,0157672799565969
25	128	0,0351186148320675	0,0602355750083602	0,0059155353276398	0,0567443031164998	0,0484023235337903	0,00798354098084109
26	256	0,0186866434899813	0,0337127236185394	0,00330053260083874	0,032705068068347	0,0391718442029867	0,00430246230926977
27	512	0,00547763127506719	0,0271478285678818	0,00191875458212788	0,0226014524778539	0,0197056901592271	0,00346802435942311
28	512 map	0,00460780470959344	0,0110104741048983	0,001673193659917	0,0214473344459635	0,0189928425001337	0,00348577612694182

## Выводы

Параллелизм дает хорошие результаты с ростом объема данных. Ускорение программы растет до определенного предела и затем перестает увеличиваться, что связано с ростом накладных расходов на создание процессов и пересылку данных.